mikro számítógép magazin

Ára: 30 Ft

KIS KARÁCSONY, NAGY KARÁCSONY...

1987/12

Az üzleti siker titka:
a jólinformáltság
és a gyors, megalapozott döntés.

**Korszerű, megbízható,
sokmunkahelyes számítógép
— fél siker!**

Gyors, pontos, megbízható!

RAIR
— SuperMicro —

**nagy kapacitású, 8—16 munkahelyes, bővíthető,
Concurrent DOS és UNIX rendszerű hálózati számítógép!**

Árában mikro — teljesítménye szuper!

Központi tára: 512 k-tól 4 M-ig
Háttértára: 50—250 MB-ig
Streamer: 45 MB
Floppyja: 1 MB vagy átkapcsolva IBM kompatibilis 360 kB
Nyelvek: BASIC, PASCAL/MT +, PL/1, C, ASSEMBLER

Rendszerszoftverek:
— DataFlex (többfelhasználós adatbázis-kezelő és adatvédelmi rendszer)
— Access Manager (többfelhasználós index-szekvenciális fájlkezelés, fájl- és rekordvédelem)
— Display Manager (képernyőkezelő szubrutinok)
— Super Sort (rendező programcsomag)

ELADÁS! GARANCIA! SZERVIZ!
LÍZING KEDVEZŐ FELTÉTELEKKEL!
SZOFTVERFEJLESZTÉS!
IRODAAUTOMATIZÁLÁSI TANÁCSADÁS!

ÖNNEK IS SZÁMÍTUNK!

SZÁMÍTÁSTECHNIKAI SZOLGÁLTATÓ KISSZÖVETKEZET
1139 Budapest, Kartács utca 27.
Tel.: 295-899, 490-778, 296-446

5. ÉVFOLYAM
1987/12. SZÁM

μ mikro számítógép magazin

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

A kiadvány a Tudományszervezési és Informatikai Intézettel együttműködve készül

A szerkesztőbizottság vezetője:
Kovács Győző

E számunkat szerkesztették:

Bakos Tamás
(programozástechnika)

Broczkó Péter
(hírek)

Kovács Győző
(levelezés)

Lindner László
(sakkprogramozás)

Petróczy Judit
(könyvek)

Simonyi Endre
(klub)

Varga András
(iskola-számítógép)

Címképünk:
Ramocsai Imri munkája

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levélcím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat

Felelős kiadó:
dr. Király G. István
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlapelőfizetési és Lapellátási Irodáján
(1900 Budapest V.,
József nádor tér 1.)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti a Kultúra,
1389 Budapest, Pf. 149.
és a Magyar Média
1932 Budapest, Pf. 279.
86—0253

Szikra Lapnyomda
Budapest (87—1670)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Tartalom

Maradék eszünkkel 2
Az új Sinclair-csoda 23
Mikrogépek, makroárak 24
Rendszerfejlesztési eszközök. Szoftver 26
Adok — veszek — cserélek 27
Miért nem stupid a MUPID? 28
Olvastunk . . . 40
μINFORM 43
Petike adót számít 45

ISKOLA-SZÁMÍTÓGÉP

Átadták az 5000. Videoton iskolaszámítógépet 3
A ROM-ban tárolt aritmetikai rutinok használata 4
TechnoMIR 6

DIÁKROVAT

Visszatekintő 8
Sprite-ok „bonckés" alatt 8
Képernyő-editor 9
Trükk Spectrumra 11

PROGRAMOZÁSTECHNIKA

BASIC és gépi kód 12
Z80 programok haladóknak Spectrumra és Primóra 13
Számítógépes grafika Pascalban. Kör 19
Függvények és utasítások 21

μPROGRAMOK

Datakonverter 31
A MASTERFILE magyar változata 32

μKLUB

Integrált szoftver 35
Rejtett fájlok 38
Óránként húszért 39

SAKKPROGRAMOZÁS

A játékfa és kiértékelése 44

AZ OLVASÓ ÍRJA 46

KÖNYVEK 47

HÍREK, ÉRDEKESSÉGEK 48

Maradék eszünkkel

„Nem javasolom a számítógépek alkalmazásának a betiltását. Magam is igen gyakran használom. De megengedhetetlen, hogy az elburjánzó számítógép-őrület maradék eszünket is elvegye."
(Idézet Andrzej Nieiec cikkéből, amely a Polityka 1986. évi 43. számában jelent meg „Megálljt kell mondani a számítógép-őrületnek!" címmel. A cikk fordítását az érdeklődők a Valóság 1987. 3. számának 122—124. oldalán olvashatják.)

Valóban megálljt kell mondani a számítógép-őrületnek?

Egy cikket keresgélve kedvenc lapom, a Valóság korábbi számait lapozgattam, amikor ismét a kezembe került Andrzej Nieiec a címben idézett cikkének fordítása. 1987 elején olvastam először az írást, és már akkor az jutott az eszembe, hogy erre válaszolnom kellene, talán éppen a Politykában. Ilyenkor – újságíró barátaimtól hallottam – az első olvasás okozta felajzott állapotot felhasználva, azonnal le kell ülni és meg kell írni a választ. Ha az ember sokat késlekedik, akkor az első érzelmi hatás már nem ismétlődik meg, a válasz nem hiteles.

Ezzel a cikkel érdekes módon másként voltam: második olvasásra is irritál, vitára kényszerít, felpiszkálja az érzelmeimet.

(Általanosítások.) Azt hiszem, semmi bajom nem volna az írással, ha a szerző azt írná, hogy véleménye szerint Lengyelországban baj van a számítógépesítési programmal, mert például egyes teoretikusok, főnökök, minisztériumok azt mondják, hogy *„Ifjúságunknak meg kell tanulnia a számítógépek programozását, ellenkező esetben a jövőben súlyos következményekkel kell számolnunk."*.

Nieiec nem ezzel a kétségkívül hamis tétellel: „számítástechnika = programozás" vitatkozik, hanem elhiszi, hogy ez a tétel igaz, és ha igaz, akkor a számítástechnikát valóban ki kell űzni az iskolából. Van egy másik példa is: „Az iskolai programok eddig egyetlen olyan feladatot sem tartalmaztak, amelyet ne lehetne papír és ceruza, legrosszabb esetben zsebszámológép segítségével megoldani.".

Amig ezt a mondatot nem olvastam, addig én sem tudtam, hogy számítógépre csak olyan feladatokat szabad vinni, amelyeket papíron ceruzával nem lehet megoldani (!?) Eszembe jutott egy korábbi eset. Az ötvenes évek végén az első elektronikus számítógépet, az M—3-at egyszer bemutattuk néhány matematikus akadémikusnak. Emlékezetem szerint a géppel trigonometrikus függvényeket, számrendszerek közötti konverziókat, egyszerű aritmetikai példákat számoltunk ki. E. akadémikus, intézeti igazgató véleményét a bemutató után röviden és velősen így foglalta össze: „Meggyőződtem arról, hogy amire itt maguk a gépet használják, azt én fél perc alatt egy csomagolópapír sarkán kiszámolom." Hallgattunk, erre nem volt mit válaszolnunk.

Én is ismerek olyan iskolákat, ahol a számítógépeket csak a tanár jelenlétében lehet használni, és csak olyan példák megoldására, amelyeket ő előre elkészített. Láttam viszont olyan esetet is, hogy egy éjjel-nappal csak számítógéppel programozó gyerek, akit hat ökörrel sem lehetett rávenni az orosz nyelv tanulására, rövid idő alatt az iskola egyik legjobb oroszosává vált, mert kiváló pedagógiai érzékű tanárának biztatására először egy cirill betűket iró, azután egy szótárt kezelő és végül egy nyelvtanító programot készített, és közben maga is megtanulta a nyelvet. Vagy ma már köztudott, hogy a számítógépekkel való kommunikáció nagyon sok gyerekben támasztotta fel az angol nyelv iránti érdeklődést és egyben a tanulási kedvet is, sokkal több gyerekben, mint például az ugyancsak angol szövegű, népszerű beatmuzsika.

Évekkel ezelőtt ott voltam azon a pályázaton is, ahol az egyik harmadikos gimnazista gyerek a Kalevala eposz fordításhűségét analizálta számítógéppel, akadémikus precizitással a szájtátó zsűri és az elbűvölt diákhallgatóság előtt. Kell-e ennél jobb propaganda a nyelvészetnek?

(Maradjunk-e a palatáblánál?) Van egy érdekes része az írásnak, amelyben az oktatástechnológiáról van szó. A cikk — röviden — azt kérdezi, hogy miért vezetnek be új eszközöket (számítógépet) az iskolákba, amikor még a régieket (vetítőgép, magnetofon stb.) sem használják az órákon, hiszen: „Az új oktatási eszközök bevezetése akkor indokolt, ha a régiek már nem látják el szerepüket, az viszont, hogy a számítógépet a XVIII. századi tananyag portalanítására használják fel, céltalan és nem gazdaságos."

A mondat második felével egyetértek, az első fele azonban számomra — konzervativ szemlélete miatt — elfogadhatatlan. Ha valóban úgy lenne, ahogyan Nieiec írta, akkor felesleges volt a palatábla helyett a füzeteket bevezetni.

Eltűnődtem. Azt hiszem, hogy ez a cikk kulcsmondata, tehát hogy *valamire, azaz az újra csak akkor van szükség, ha a régit már nem lehet használni.*

Majdnem feleslegesnek érzem ezek után az olyan érvelést, hogy

— az oktatástechnológia korszerű eszközei, mint például a számítógép is, ha jól alkalmazzák, akkor nagymértékben megnövelheti a tanulás *hatékonyságát*, vagy

— *átvállalja* a tanár néhány rutinfeladatát, mint például a tananyag ismétlése, az egyes anyagrészek kikérdezése stb., és így

— lehetőséget és *időt ad* a tanárnak, hogy pedagógiailag, módszertanilag, de tartalmilag is olyan tananyagokat készítsen, amelyek megkönnyítik a tanulást, és így a diák terhelését nagymértékben csökkentik.

(Egy árulkodó idézet.) „Az oktatóprogramok többségének több súlyos fogyatékossága van, mint amennyi kétes előnye. A számítógépes számtanórák egyszerű játékok, amelyekkel a *gyermek saját akaratából több órát úgy tölt, hogy figyelmét maximálisan megfeszíti.* Ha sikerülne ilyen hosszú időre felkelteni a diák érdeklődését egy olyan tanóra iránt, melyet pedagógus tart, az eredmény sokkal jobb lenne." Erre van egy magyar mondás: „Ha Mari néninek négy kereke lenne, akkor autóbusznak neveznék". Nieiec voltaképpen ebben a kijelentésében két nagyon fontos dolgot mond ki anélkül, hogy abból a szükséges következtetést levonná. Nevezetesen:

— a számítógépes számtanórákon (hozzátesszem: az otthoni számítógép mellett is, a szakkörökön is és mindenhol, ahol a számítógéphez közel kerülhet) a gyerek *saját akaratából* több órát képes eltölteni, illetve

— hogy általában *nem szokott sikerülni a diák figyelmét felkelteni* hasonlóan hosszú időre egy pedagógus által vezetett tanóra iránt. Ha néhány középiskolai tanáromra gondolok: Béla Pálra, Létay Menyhértre, Pataki Józsefre vagy egyetemi professzoraimra: Simonyi Károlyra, Kozma Lászlóra és másokra, *akkor ez sem igaz!* Márpedig ha ez szerinte így van, akkor ennél jobb érv nem kell a számítógép oktatási alkalmazására, mert ha jól emlékszem, tanulni úgy kell, hogy az ember — sokszor hosszú ideig — ismételgeti az anyagot, egészen addig, amíg meg nem tanulta. És minden olyan eszközt, ami a tanulás iránti figyelmet hosszú időn keresztül fenn tudja tartani, alkalmazni kell, függetlenül attól, hogy az számítógép, könyv, video vagy akármi más.

Nem vitatkoznék a szerzővel, ha nem a számítástechnika, a számítógép *ellen*, hanem annak okos, hasznos, hatékony alkalmazásáért szólna. Ezért nem tudok egyetérteni olyan megállapításaival, hogy „a számítógépes játék nyilvánvalóan tompítja a diákok eszét" és hogy „legnépszerűbbek a csihi-puhi játékok (. . .), kevésbé népszerűek a szimulációs játékok (. . .), a kalandjátékokra pedig kevés a vállalkozó, mivel a fiatalok nem ismerik az idegen nyelveket . . .". Az én tapasztalatom merőben más. Szerintem a gyerekek jó — talán nagyobbik — részét egyre kevésbé érdeklik az ügyességi, népszerű nevén lövöldözős programok, sokkal jobban elszórakoznak a szimulációs, kaland- és logikai játékokkal.

Nieiec cikkében persze nemcsak vitatható állítások vannak. Jó néhány megállapításával teljes egészében egyetértek. Például azzal is, hogy „Az informatikát kétségkívül oktatni kell, de nem mindenkinek: csak azoknak, akik ezzel kívánnak foglalkozni, és nem a tanórák keretében, hanem szakkörökben vagy tagozatos osztályokban.".

Egy másik idézet: „Ahelyett, hogy az ifjúságot a számítógépek programozásával gyötörnék, jobb lenne a fiatalok lelkesedését a munkaszervezési elvek iránt felkelteni, ami a társadalom informatizálásának az egyik alapja."

Már terjedelmi okokból sem sorolom fel a cikk minden pozitív és negatív megállapítását. Ha sikerült a mondanivalómat szándékom szerint megfogalmaznom, akkor az olvasók bizonyára megértették, hogy az egyes rossz példákból levont általánosítások jogosultságát vitatom. Azért, mert valakik valamit rosszul csináltak, azért azt a valamit még jól is meg lehet csinálni. A számítógép életünkben nem szükséges rossz, amivel kénytelenek vagyunk együttélni, hanem olyan eszköz, amelyet meg kell ismerni, és ha már megismerték, akkor biztosan meg fogják szeretni, ha pedig szeretik, akkor majd jól alkalmazzák. Az autót sem lehet azért kidobni, mert elűti a tyúkokat, és a repülést sem kell abbahagyni, mert néhány repülőszerencsétlenség sok száz ember értelmetlen halálát okozta.

Nieiec cikkét, ha jól belegondolok, még hasznosnak is tartom, hiszen az embert vitára és bizonyos helyzetek, módszerek újragondolására készteti. Nem hiszem, hogy egy írás bárhol is véget vetne a „számítógép-őrületnek", és a számítógépek otthoni, iskolai vagy munkahelyi terjedésének például Lengyelországban vagy nálunk útját lehetne állni. A számítástechnika és a számítógépek — véleményem szerint — egyáltalán „nem veszik el maradék eszünket". Ellenkezőleg: azt a kicsit is, ami van, intenzív gondolkodásra serkentik.

Valamennyi olvasónknak kellemes ünnepeket kívánok!

KOVÁCS GYŐZŐ

Átadták az 5000. Videoton iskola-számítógépet

Köpeczi Béla művelődési miniszter átadja a Sámsonkerti Általános Iskola igazgatónőjének az 5000. Videoton iskolaszámítógépet

Sötétkék ruhás, fehér inges, piros nyakkendős gyerekek sorakoznak a Sámsonkerti Általános Iskola udvarán. Ünnep ez a nap, hiszen a kis tanyasi iskolába ellátogatott Köpeczi Béla művelődési miniszter, hogy átadja az ötezredik Videoton iskola-számítógépet.

Az iskola a harmincas években épült. Ma négy tanteremben oktatnak. A településen, Sámsonkerten 350 család lakik, az iskolában 72 gyerek tanul. A tanyaközpont Debrecentől 14, Hajdúsámsontól 6 kilométerre terül el. Az oktatási feltételek bizony nem ideálisak. Osztatlan rendszerben tanítanak, vagyis az I—III., II—IV. és az V—VI., VII—VIII. osztályokat összevonták. Az osztatlan torna gondjai mellett dicséretes nevezetessége is van a sámsonkerti iskolának: a hattagú tantestület 1977-ben Állami dijat kapott! Valamennyien nők, s Debrecenből járnak a tanyaközpontba. Lelkesedésük, pedagógiai gyakorlatuk, szakmai tudásuk garancia lehet arra, hogy a sámsonkerti gyerekek mindazt megtanulják, amit ma az általános iskolás diákoknak tudniuk kell.

Dr. Andó Lászlóné igazgatónővel és Hramura Gábornéval, a tagiskola igazgatójával beszélgetve mindjárt a számítógépes szaktantermükbe invitálnak. A kis szobában úttörőjelvények diszelegnek a falon — egyben ez a helyiség az úttörőszoba is —, az asztalon viszont ott áll a számítógép és a televízió.

— Ősszel kaptunk egy Videoton gépet, nyáron dr. Szegedi Sándorné — a matematika-fizika szakos tanárunk — elvégezte a megyei pedagógiai intézet szervezte tanfolyamot, így ebben a tanévben már megismerhették a gyerekeink a számítógépet. Amikor a tanárnőnek ideje volt, foglalkozott az érdeklődő gyerekekkel, akik nagyon figyeltek, és indulásként néhány dolgot meg is tanultak. Sőt, télen Zánkán több tanulónk részt vett a számítógépes táborban. A felsősöket küldtük, hiszen náluk tapasztalható elsősorban a számítástechnika iránt érdeklődés.

Jócsák István hetedikes bizonyult eddig a legügyesebbnek.

— Délutánonként főleg játszottunk, amikor a tanárnő ráért. A gépet tudjuk kezelni, sőt Homlok Tibivel és Bordás Attilával már megpróbáltuk programozni is. Sajnos kevés a kész programunk.

— Örömünkre — mondja Andó Lászlóné — most egyszerre nem egy, hanem mindjárt két Videoton gépet kapunk, ugyanis a megyei tanácstól is érkezett számítógép. Ez lehetővé teszi számunkra, hogy a közeljövőben szakkört szervezzünk.

Gépkocsik sora tűnik elő a homokos dűlőúton. Megérkezik a miniszter, a megyei és a Debrecen Városi Tanács művelődési osztályvezetőinek kíséretében, és eljött Párizs György minisztériumi főtanácsos, a Tudományszervezési és Informatikai Intézet igazgatója, valamint Gantner János, a Videoton Számítástechnikai Gyárának igazgatója is. Az igazgatónő az egyik tanterembe kíséri vendégeit.

A köszöntő szavak után Köpeczi Béla művelődési miniszter mond beszédet.

— Örömmel és szeretettel köszöntöm a Sámsonkerti Általános Iskola pedagógusait és diákjait az ötezredik Videoton gyártmányú iskolaszámítógép átadása alkalmából. Büszkén állíthatjuk, hogy a magyar közoktatás lépést tartva a legkorszerűbb oktatási rendszerekkel — a szerény lehetőségekhez mérten — bár nagy anyagi áldozatokkal, de követni tudta a jobb gazdasági helyzetben lévő országok számítástechnikai okatásának fejlődési ütemét. Ezt bizonyítja, hogy az iskolák 1983-ban kapták az első számítógépeket, és ma, négy év után már csaknem huszonötezer gép működik: ezek közül mintegy tízezren az általános iskolások tanulhatnak.

Ma már kétségtelen, hogy a legkülönfélébb szakmák, tudományok, de a mindennapok sem nélkülözhetik a számítástechnikai ismereteket. És az is nyilvánvaló mindannyiunk számára, hogy ezeket a korszerű, a fejlődés, a haladás szempontjából nélkülözhetetlen ismereteket csak akkor szerezhetjük meg, ha ehhez a megfelelő eszközök léteznek. Sok tudományban lehet haladni, ha adott az érdeklődés, a szorgalom, a tehetség, de egy ilyen, technikához kötött tudást csak akkor szerezhetünk meg, ha az itt sem nélkülözhető érdeklődés, szorgalom, tehetség mellé a szükséges gépek és felszerelések is megvannak.

Szeretném jelképnek tekinteni, hogy ennek az ünnepélyes pillanatnak egy kis település a helyszíne. Jelkép abból a szempontból, hogy nincs olyan kicsiny helység az országban, ahová ne kellene eljuttatni a modern technikát nevelésünk, oktatásunk korszerűsítésére. Magyarországnak jó alapműveltségű, felkészült szakmai tudású, önmagukat becsülő és népüket szerető hazafiakra van szüksége, akik tudnak alkalmazkodni az új körülményekhez, alakítják is azokat. E cél megvalósítását kívánva adom át a Sámsonkerti Általános Iskolának az ötezredik Videoton gyártmányú iskolaszámítógépet.

Dr. Szekeres Antal, a megyei tanács művelődési osztályának vezetője ezt követően jelenti be, hogy a megyei tanács is ad egy Videoton számítógépet, a Videoton gyár pedig egy szines televíziót ajándékoz a sámsonkerti iskolásoknak. S hogy az öröm még teljesebb legyen, Párizs György programokat hozott magával, és azokat átadja az iskola igazgatónőjének.

Mit jelent ez a nap e kis település iskolájának életében? Elsősorban azt, hogy a jövőben lehetőség nyilik arra, hogy a gyerekek szervezetten, szakköri formában tanuljanak meg a számítógépekkel bánni, és most már ne csak játsszanak azokkal, hanem fontosságának megfelelően alkalmazzák a technikát. Hramura Gáborné arról álmodozik, hogy kollégái hamarosan a tanítási órákon is használják majd a számítógépeket.

Pillanatokon belül a szaktanterembe kerül a két gép. Hajdú-Bihar megyében ezután már 1500 számítógépet tarthatnak nyilván az iskolák. És ez nem kis szám. A számítógépes program a megyében megfelelő ütemben folytatódik . . .

FILEP TIBOR

A ROM-ban tárolt aritmetikai rutinok használata

TVC

Előző számunkban bemutattuk a TV-Computer ROM-jának adatmozgató szubrutinjait. A gép az adatmozgató rutinokhoz hasonlóan a négy alapműveletet is a veremben elhelyezett számok között végzi el a következő módon.

A veremben egymás fölé kell tenni a két operandust (OP1, OP2), amelyek között elvégzi a műveletet. Az eredmény az elsőnek betett (OP1) szám helyén keletkezik, az OP2 pedig elvész (definiálatlan).

A szubrutinok hívása az előzőekben már ismertetett módon, az RST 18H utasítást követően, a szubrutin sorszámának megadásával történhet. Ebben az esetben is igaz, hogy ha a sorszám legnagyobb helyiértékű (128-as súlyú) bitje 1, akkor ez az utolsó végrehajtandó szubrutin sorszáma; az ezt követő kódot a processzor már utasításként értelmezi és végrehajtja.

Az *1. táblázat* a szubrutinokat mutatja be. IY a veremmutató, a verem pedig a memóriában fentről lefelé terjed. Az utolsó sorban a 04-es sorszámú szubrutin a veremben levő szám előjelét változtatja ellenkezőjére.

Szubrutin hívása	OP1 helye	OP2 helye	Művelet	Eredmény helye
RST 18H 00H	IY	IY-9	+	IY
RST 18H 01H	IY	IY-9	/	IY
RST 18H 02H	IY	IY-9	*	IY
RST 18H 03H	IY	IY-9	-	IY
RST 18H 04H	IY	nincs	CHS(X)	IY

1. táblázat

A gép beépített BASIC függvénykészlete hívható a felhasználói gépi kódú programokból is, bár itt meg kell jegyezni, hogy e függvények használatával a gépi kódú programozás legfőbb előnye, a futási idő jelentős csökkenése nem érvényesül.

Ez a probléma a beépített függvények kikerülésével oldható meg úgy, hogy a nem használt függvényt a felhasználó által megírt szubrutinnal pótoljuk. Természetesen ehhez gyorsabb algoritmusra van szükség.

A gépi kódú felhasználói függvények a ROM ismertetett adatmozgató és aritmetikai szubrutinjainak alkalmazásával készülhetnek, de ha a futási idő nem elsődleges, akkor minden további nélkül hívhatják a *2. táblázatban* szereplő függvényeket. Hívás előtt az argumentumot a verembe kell helyezni, az eredmény ugyanitt keletkezik.

Függvény hívása	A függvény BASIC megfelelője
CALL EAECH	ABS(X)
CALL EBA7H	COS(X)
CALL EBB8H	EXP(X)
CALL ECAFH	INT(X)
CALL ED21H	LOG(X)
CALL EDE3H	PI
CALL EE3EH	SGN(X)
CALL EE5EH	SIN(X)
CALL EECAH	SQR(X)
CALL EFFBH	TAN(X)

2. táblázat

A *3. táblázatban* összefoglalásképpen megadjuk az RST 18H utáni sorszámokhoz tartozó szubrutinok memóriabeli címét, így azok CALL-lal gyorsabban hívhatók. Ekkor a program hossza nő. A „!"-et csak az RST 18H-val célszerű hívni.

3. táblázat

RST 18H utáni sorszám	CALL-lal hívható cím	Funkció	Megjegyzés
00H	F493H	OP1=OP1+OP2	IY=IY+9
01H	F5FBH	OP1=OP1/OP2	IY=IY+9
02H	F512H	OP1=OP1*OP2	IY=IY+9
03H	F48EH	OP1=OP1-OP2	IY=IY+9
04H	F726H	OP1=CHS(OP1)	IY=IY
05H	EA82H	(IY)=N. KONST.	IY=IY-9
06H	EA9FH	(IY)=X reg.	IY=IY-9
07H	EA9AH	(IY)=Y reg.	IY=IY-9
08H	EAD2H	X reg.=(IY)	IY=IY
09H	EACDH	Y reg.=(IY)	IY=IY
0AH	EAC3H	X reg.=(IY)	IY=IY+9
0BH	EABEH	Y reg.=(IY)	IY=IY+9
0CH	FA92H	(IY-9)=(IY)	IY=IY-9
0DH	FB28H	(HL)=(IY)	IY=IY+9
0EH	EA68H	Lásd gépi kódú felhasználói függvények.	

A függvényeket megvalósító gépi kódú rutinok szerkezete:

— az első két bájt az előző BASIC válto-

```
7000 (1B58H) ..                    ←PEEK(5924)
             ..                    ←PEEK(5925)
             02                    függvény nevének a hossza
             53                    'S'
             48                    ,H,
             0A                    típusbájt
             CD,68,EA (CALL EA68)  változó a verembe (IY)=A
             CD,B8,EB (CALL EBB8)  (IY)=EXP(A)
             DF       (RST 18)
             08                    Xreg.=EXP(A)
             05
             01                    (IY-9)=1
             06                    (IY-18)=EXP(A)
             01                    (IY-9)=1/EXP(A)
             03                    (IY)=EXP(A)-1/EXP(A)
             05
             01                    (IY-9)=1
             0C                    (IY-18)=1
             00                    (IY-9)=2
             81                    (IY)=(EXP(A)-1/EXP(A))/2
             C9       (RET)
```

1. lista

```
7000 (1B58H) ..
             ..
             05
             50                    'P'
             42                    R
             4F                    O
             42                    B
             41                    ,A,
             0A
                LD   A, 96
                CALL  FD54
             ..... CALL  F0A7
N-db változó    LD   A, A4
esetén N-1-szer CALL  FD54
alkalmazni ..... CALL  F0A7
                LD   A, 95
                CALL  FD54
                RST  18
             0C
             02
             0A
             0C
             02
             06
             80
                RET
                                   PROBA ( X , Y )
```

2. lista

3. lista

```
10 LOMEM 8000
20 FOR I=0 TO 24
30 READ A: POKE 7000+I,A:NEXT
40 POKE 7000,PEEK(5924):POKE 7001,PEEK(5925)
50 POKE 5924,88: POKE 5925,27
60 INPUT PROMPT"X=?":A
70 PRINT SH(A):GOTO 60
80 DATA 0,0,2,83,72,10,205,104,234,
        205,184,235,223,8,5,1,6,1,3,
        5,1,12,0,129,201
```

zóra vagy függvényre mutató cím, ezt a függvény memóriabeli elhelyezése után POKE utasításokkal kell beállítani,

— a függvény nevének hossza (1 bájt),

— a függvény nevét alkotó ASCII karakterek,

— típusbájt, amely jelen esetben 0AH (10),

— a függvény argumentumát a verembe elhelyező rutin, amely a következő lehet: egy bemenő paraméter esetén CALL EA68H *(1. lista);* több bemenő paraméter esetén a *2. lista* szerinti programrészletet kell alkalmazni, N darab változó esetén N−1-szer,

— maga a függvényt megvalósító programszegmens,

— Return.

Az így megalkotott felhasználói függvényt a BASIC beépített függvényeihez hasonlóan a nevével és a numerikus paraméter(ek) megadásával lehet hívni.

A függvényeket a LOMEM utasítással védett területen célszerű elhelyezni. Egy függvény esetén a gyári leírásban szereplő CHAIN rendszerváltozó értékét kell POKE utasításokkal elhelyezni a függvény első két bájtjában, az első bájt címét (kezdőcím) pedig be kell tenni a CHAIN változóba. A CHAIN az 5924, 5925-ös decimális címen van.

Több függvény fejlécét úgy kell összeláncolni, hogy az mindig az előző függvény kezdőcímét tartalmazza. A CHAIN értékét az első függvény fejlécébe kell bevinni, a CHAIN pedig az utolsó függvény kezdőcímét tartalmazza.

Az 1. listán látható függvénynek egy bemenő paramétere van, és azt az EA68H címen lévő szubrutin teszi a verembe. A program az $(EXP(X)-EXP(-X))/2$ függvényt valósítja meg, és SH(X) alakban hívható. A hozzá tartozó BASIC programrészletet a *3. lista* tartalmazza.

A 2. lista a két bemenő paraméteres függvény felépítését mutatja (PROBA(X,Y)), amelyben nem az EA68H szubrutin teszi a változókat a verembe, hanem a megjelölt programrészlet, amelyet annyiszor kell alkalmazni, ahány vesszővel vannak elválasztva a változók. A nyitó zárójel kódja a hexadecimális 96, a vesszőé az A4, a bezáró zárójel kódja pedig a 95. A programrészlet az $SQR(X^2+Y^2)$ függvényt számolja ki (komplex szám abszolút értéke). Ez a függvény a *4. lista* BASIC programjával hozható létre.

BOTTYÁN ZOLTÁN

4. lista

```
10 LOMEM 8000
20 FOR I=0 TO 41
30 READ A: POKE 7000+I,A: NEXT I
40 POKE 7000,PEEK(5924):  POKE 7001,PEEK(5925)
50 POKE 5924,88 :POKE 5925,27
60 INPUT PROMPT"A , B =?":EGY,KETTO
70 PRINT PROBA(EGY,KETTO) :GOTO 60
80 DATA 0,0,5,80,82,79,66,65,10,
        62,150,205,84,253,205,167,240,
        62,164,205,84,253,205,167,240,
        62,149,205,84,253,223,12,2,10,
        12,2,6,128,205,202,238,201
```

TechnoMIR

A DIN modul

A „digitális input" modul a TechnoMIR interfészcsalád egyik alapáramköre. Mint a neve is elárulja, digitális jelek fogadására alkalmas. A modul továbbítja a számítógép felé a bemenetén megjelenő egybájtos TTL (0 V vagy 5 V) szintű adatokat, melyeket a 8 darab bemenetre kapcsolható záró érintkezők vagy ellenállásukat változtató passzív érzékelők szolgáltathatnak. A modul tehát képes egyszerű kapcsolók állapotát érzékelni, például ajtók, ablakok, mozgó alkatrészek helyzetét lekérdezni. Fotoellenállás, termisztor és egyéb egyszerű jelátalakítók segítségével, amelyek a környezet nem villamos jellegű mennyiségeit villamos jellé alakítják, a DIN modult felhasználva, számítógéppel ellenőrizhetjük a környezet bizonyos állapotváltozásait. Természetesen ezt csak úgy, hogy jelzi: van-e fény vagy nincs, elértünk-e valamely hőfokot vagy még nem, zárva van-e a kapcsoló vagy nincs.

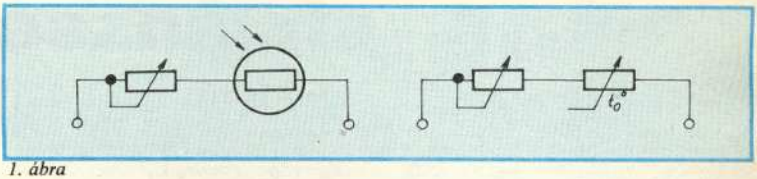
Várható probléma, hogy az átalakítók érzékelési szintjét a széles mérési tartomány érdekében meg kell változtatni. Erre az *1. ábrán* mutatunk be egy megoldást. A termisztorhoz, illetve a fotoellenálláshoz kapcsolódó változtatható értékű ellenállással (potenciométerrel) az érzékelési szint állítható be.

A módszer használata során nem szabad figyelmen kívül hagyni az átalakítók nemlineáris átalakítási karakterisztikáját. A potenciométer ellenállásának hőmérsékletfüggéséből adódó hibát úgy védhetjük ki, hogy minél távolabb helyezzük el az érzékelőtől, a hőmérsékletváltozás érzékelésének helyétől.

Fontos tudnivaló, hogy a modul a bemenetére kapcsolt adatokat nem tárolja, azok pillanatnyi értékét kérdezhetjük le megfelelő utasítással.

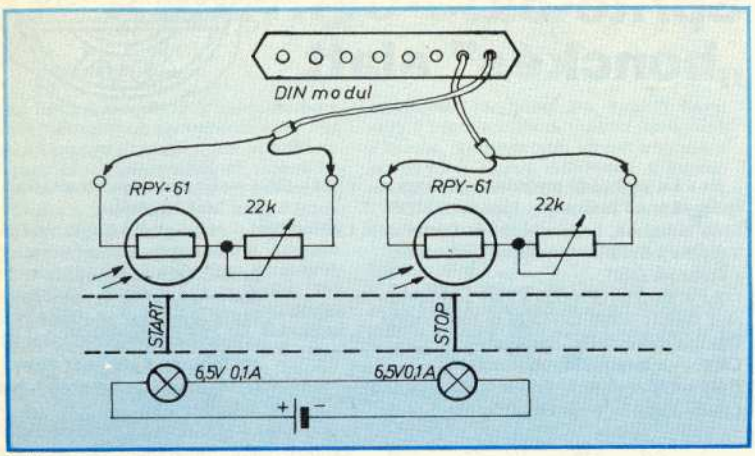
A DIN modul címe a HT gépen: 24,
tehát a HT iskolaszámítógépnél: INP(24),
a Primónál: INP(216),
a C64-nél: PEEK(56856),
a C Plus/4-nél és C16-nál: PEEK(65048) utasításokkal tehetjük meg. Ha a bemenetre nem kapcsolódik mérőzsinór, alaphelyzetben a modul 0 szintet érzékel. Ezt egy rövid tesztprogrammal ellenőrizhetjük:

```
10 A=INP(24)
20 PRINT A
```



1. ábra

```
10 REM ***********************************
20 REM *                                 *
30 REM *         OPTOKAPUS MERES         *
35 REM * TechnoMIR INTERFACE SOFTWARE *
40 REM *                                 *
45 REM ***********************************
50 CLEAR 1000:CLS:GOSUB 450:CLS
60 PRINT@64*7+10,'HANY VERSENYZO INDUL(max.30)';
65 INPUT N:N=INT(N)
70 IF N>1 OR N<30 THEN 90
75 PRINT:PRINT'          ROSSZ!'
80 FOR K=0 TO 1000:NEXT K:CLS:GOTO 60
90 DIM N$(N),E(N),E1(N),S(N):GOSUB 400
100 CLS:FOR B=1 TO N
    ***********************************
    *      N DARAB IDO MERESE         *
    ***********************************
110 PRINT@64*7+22,N$(B);' INDULHAT!'
120 Z=INP(24):IF Z=3 THEN 120
130 IF Z=2 THEN CLS:GOTO 140 ELSE GOTO 120
140 E(B)=0
150 E(B)=E(B)+.1:FOR X=1 TO 5
160 IF INP(24)=1 THEN 170 ELSE NEXT X:GOTO 150
170 PRINT:PRINT'      IDO :';E(B)
180 FOR X=1 TO 300:NEXT X:NEXT B:CLS
    ***********************************
    *        SORBARENDEZES            *
    ***********************************
190 H=0:FOR A=1 TO N
200 E1(A)=E(A):NEXT A:H=1
210 FOR B=1 TO N:FOR A=1 TO N
220 IF E(A)<=E(H) THEN H=A
230 NEXT A:S(B)=H:E(H)=100000:NEXT B
    ***********************************
    *          KIIRATAS               *
    ***********************************
300 A=0:GOSUB 380
310 PRINT:A=A+1:B=S(A)
320 C=A-12*INT(A/12)+3:C=C*64
330 IF A=12*INT(A/12) THEN GOSUB 390:GOSUB 380
340 PRINT@C,'* ';A;'.  ';N$(B);' (';B;')';
345 PRINT TAB(50);E1(B)
350 IF A=N THEN PRINT' ENNYI':GOSUB 390:END
360 GOTO 310
```



2. ábra

```
    ************************************
    *          SUBRUTINOK              *
    ************************************
380 PRINT TAB(17)'E R E D M E N Y L I S T A'
385 RETURN
390 IF INKEY$='' THEN GOTO 390
395 RETURN
400 CLS:FOR A=1 TO N:PRINT A;'.';
410 INPUT 'VERSENYZO NEVE';N$(A)
430 NEXT A:RETURN
450 CLS:PRINT:PRINT
460 PRINT' Ez a program egy verseny
 idoeredmenyeinek meresere alkalmas.'
470 PRINT'
480 PRINT' Ket fotocella van. A program
 a ketto kozotti athaladas idejet'
490 PRINT' meri. Ha az elso fotocella
 elott athalad a versenyzo
500 PRINT' az ido merese. Az ora akkor
 all meg, ha a masodik fotocella elott'
510 PRINT' is athaladt.'
520 PRINT'      MEG AKAROD VIZSGALNI A
 FOTOCELLAKAT (I/N)?'
530 IF INKEY$='I' OR INKEY$='i' THEN 550
540 RETURN
550 CLS:PRINT:PRINT
560 PRINT ' Tedd a kezed az egyik fotocella
 ele, es figyeld a kiirt szamot!'
570 PRINT ' Ha az 1 vagy 2, akkor a potencio
 meterek be vannak allitva. Ha mas'
580 PRINT ' szamot latsz, akkor allitsd be!'
590 PRINT:PRINT 'Ha vegeztel, nyomd meg a V-t!'
600 PRINT@14*64+30,INP(24)
610 IF INKEY$='V' OR INKEY$='v' THEN RETURN
620 GOTO 600
```

30 GOTO 10

Ha valamelyik bemenetre mérőzsinórt csatlakoztatunk és összekapcsoljuk a két banándugót, akkor az adott bemenet helyiértékének megfelelő szám jelenik meg folyamatosan a képernyőn. A modul iskolai alkalmazása során igen előnyös lehet, hogy a program futása közben nyugodtan kihúzhatjuk a mérőzsinórt vagy újabbat csatlakoztathatunk a modulhoz.

Előfordulhat, hogy egyszerre több bemenetet használunk érzékelésre, de egy adott pillanatban csak az egyik bemeneten levő szintet kell figyelni. Ilyenkor a maszkolásnak nevezett programozástechnikai trükkel kiküszöbölhetjük azt a problémát, hogy a modul bemenetén megjelenő egybájtos adat decimális értékében mind a 8 bemeneten levő adat benne van. Például a

10 IF (INP(24)AND4)=4 THEN 20

programsorral elérhetjük, hogy a számítógép csak a DIN modulból beolvasott adat 3. bit értékét figyeli, függetlenül a többi bemeneten lévő adattól. A példánkban csak akkor ugorna a 20-as sorra a számítógép, ha a 3. bemenet logikai „1" szintet érzékelne.

Bemutatunk egy példát a DIN modul gyakorlati alkalmazására.

Optokapus mérés. Ebben a példánkban a modul széles körű felhasználhatóságát kívánjuk bemutatni. A közölt lista alapján elkészülő programot akár testnevelési órán, akár fizikaórán használhatjuk. Magát a programot, amely HT1080Z számítógépre született, a modul rövid ismertetése után Juhos Botond és Horváth Zoltán első osztályos tanulók készítették a Bolyai János Híradástechnikai Szakközépiskolában.

A *2. ábrán* látható a méréshez szükséges periféria elrendezési rajza. A perifériában a DIN modulon kívül 2 db optokapu található. Az optokapu fotoellenállásból és valamilyen fényforrásból áll. A program elején ennek az érzékenységét egy rövid tesztprogram segítségével beállíthatjuk. A program a két optokapu előtt való elhaladás közben eltelt időt méri. Tehát az első kapu a START, a második a STOP. Vigyázat! A program nem valós időt mér, hanem egy ciklusváltozót számlál, amelyben a ciklusidő közel egy másodperc, de persze ez is alkalmas arra, hogy eldöntse a versenyzők közötti sorrendet.

A pontos időmérésre a HCLK (hardware clock) modul kapcsán fogunk visszatérni.

ALBU LÁSZLÓ—KIRÁLY LÁSZLÓ

Visszatekintő

A téli napokon jó visszagondolni a még oly szeszélyes nyárra is, a talán most sem késő a programban gazdag kánikulai emlékek felidézése. Aki nem volt ott, annak a jövőre ismét megrendezendő táborhoz előzetes kedvcsináló lehet, aki meg az alábbi történet egyik szereplője volt, annak kiváló alkalom a nosztalgiázáshoz.

A nyáron számítástechnikai, elektronikai és gépjármű-technikai tábort rendezett a KISZ KB Középiskolai és Szakmunkástanuló Tanácsa. Az egyhetes program résztvevőinek a tiszafüredi Expressz tábor adott otthont, a számítógépes kurzusoknak pedig a Kiss Pál Általános Iskola.

Tiszafüreden a táborban meglepetésként számítógépes adatnyilvántartó rendszer — adataikat feldolgozva — várta a nagyérdemű társaságot. Az első nap az ismerkedés jegyében telt, majd este diszkóprogram következett. Sajnos a fiú-lány arány meglehetősen rossz volt, mert csak minden tizedik fiúnak jutott egy-egy táncos lábú leányzó. A hoppon maradtaknak azért vigasztalásul szolgálhatott, hogy tulajdonképpen nem táncolni utaztak a Tisza-partra, hanem azért, hogy a számítógépet bűvöljék. A következő napokban erre számtalan lehetőség kínálkozott . . .

Reggel fél hétkor kollektív ébresztés, melyet bőséges reggeli követett. Busszal utaztak a majd' két és fél kilométerre lévő iskolába, „második otthonukba". Ott lendült csak be az igazi élet, mindenkinek tudása, érdeklődése szerinti fokon: kezdő és haladó BASIC, PASCAL, Z80 ASSEMBLER, 6510 ASSEMBLER, FORTH és C nyelveken. Ugyanitt olyan lengyel számítógépekkel találkoztak, amelyeken angol nyelvtudásukat gyakorolhatták. Természetesen a mikroszámítógép-parkból nem hiányoztak a legkülönbözőbb típusok.A gyerekek dolgozhattak az IBM PC/AT, Primo, HT 1080Z, TVC, ENTERPRISE, ZX—SPECTRUM, C+4, C16 és C64 gépeken. Egyébként ez utóbbiból volt a legtöbb. Sajnos a lemezegységekből és nyomtatókból már nem dicsekedhettek ekkora bőséggel. Munka közben, háttérzajként számítógépes zenét hallgattak, s a gépeket este kilencig pülölték. A mohóbbja a hedonista élvezeteket kihagyva, még az étkezési idő alatt sem vált meg a mikroszámítógépétől.

Azért senki ne gondolja, hogy szórakozásra nem jutott energia. Kánikulát fogott ki a táborozó sereg, a sátrak nappal izzottak, éjszaka viszont ontották a hideget . . . és ehhez elemi csapásként nem került el senkit a szúnyoginvázió, meg a tiszai fürdőzés gyönyörűsége. Vetélkedők, videózás és bogrács-partik gazdagították a táborozók napjait. No meg a takarodó utáni beszélgetések, vitatkozások, melyek gyakran egészen az ébresztőig eltartottak. Természetesen mindig ugyanarról folyt a téma, de azt rendesen körbejárták. A fanatikus számítógép-rajongók a mikrogépekről társalogtak.

E szűkre szabott idő mérlege végül is a következő: a hasonló érdeklődési körű gyerekek összebarátkoztak, sokat tanultak, gyakoroltak, és végül is vakáció alatt kedvtelésüknek hódolhattak.

ÉNEKES FERENC
és a Diákszerkesztőség

Sprite-ok „bonckés" alatt

DIÁK Commodore ROVAT

Ez a kis gépi kódú program segít, hogy a sprite-okat ne csak a gép kirajzolta képernyőn láthassuk, hanem alatta és felette is. Vagyis a keret bizonyos részét „lebontjuk" a sprite-ok fölül.

Az *1. ábra* tartalmazza a gépi kódú program listáját. C000-tól C020-ig tart a megszakítás előkészítése. Ezen belül:

C000 — a megszakítások letiltása
C001—C00A — az új vektor betöltése
C00B—C014 — a raszter értékének 0-ra állítása
C015—C01F — a raszter szerinti megszakítás engedélyezése
A megszakítások által működtetett, ismétlő programrész
C021-től C045-ig tart
C021—C026 — a raszter szerinti megszakítást kérő bit törlése
C027—C045 — a program két ágra szakad. A 0-ás és 249-es rasztersoroknál bekövetkező megszakítás a 24 illetve a 25 soros üzemmódokat váltja egymással és a következő raszterértéket tölti be. Majd végül folytatja a gép saját megszakítási rutinját, vagy visszatér a megszakításból.

A *2. ábra* a BASIC-betöltőt tartalmazza. Akiknek monitorprogramjuk van, azok a *3. ábrán* található hexadecimális lista alapján valamivel gyorsabban üthetik be programjukat.

GYURICZA PÉTER

1. ábra

```
.; C000  78       SEI
.; C001  A9 21    LDA #$21
.; C003  A0 C0    LDY #$C0
.; C005  8D 14 03 STA $0314
.; C008  8C 15 03 STY $0315
.; C00B  A9 00    LDA #$00
.; C00D  A0 1B    LDY #$1B
.; C00F  8D 12 D0 STA $D012
.; C012  8C 11 D0 STY $D011
.; C015  A9 81    LDA #$81
.; C017  A0 7F    LDY #$7F
.; C019  8D 1A D0 STA $D01A
.; C01C  8C 0D DC STY $DC0D
.; C01F  58       CLI
.; C020  60       RTS
.; C021  AD 19 D0 LDA $D019
.; C024  8D 19 D0 STA $D019
.; C027  AD 12 D0 LDA $D012
.; C02A  D0 0D    BNE $C039
.; C02C  A9 F9    LDA #$F9
.; C02E  A0 1B    LDY #$1B
.; C030  8D 12 D0 STA $D012
.; C033  8C 11 D0 STY $D011
.; C036  4C 31 EA JMP $EA31
.; C039  A9 00    LDA #$00
.; C03B  A0 13    LDY #$13
.; C03D  8D 12 D0 STA $D012
.; C040  8C 11 D0 STY $D011
.; C043  4C BC FE JMP $FEBC
.; C046  00       BRK
```

2. ábra

```
10 FORI=0TO69:READX:POKE49152+I,X:S=S+X
15 NEXT
20 IFS=8301THEN40
30 PRINT"HIBA A DATA SOROKBAN":END
40 PRINT"OK.":SYS49152
50 POKE53280,14:POKE53281,0
100 DATA120,169, 33,160,192,141, 20,  3
110 DATA140, 21,  3,169,  0,160, 27,141
120 DATA 18,208,140, 17,208,169,129,160
130 DATA127,141, 26,208,140, 13,220, 88
140 DATA 96,173, 25,208,141, 25,208,173
150 DATA 18,208,208, 13,169,249,160, 27
160 DATA141, 18,208,140, 17,208, 76, 49
170 DATA234,169,  0,160, 19,141, 18,208
180 DATA140, 17,208, 76,188,254
```

3. ábra

```
.:C000 78 A9 21 A0 C0 8D 14 03
.:C008 8C 15 03 A9 00 A0 1B 8D
.:C010 12 D0 8C 11 D0 A9 81 A0
.:C018 7F 8D 1A D0 8C 0D DC 58
.:C020 60 AD 19 D0 8D 19 D0 AD
.:C028 12 D0 D0 0D A9 F9 A0 1B
.:C030 8D 12 D0 8C 11 D0 4C 31
.:C038 EA A9 00 A0 13 8D 12 D0
.:C040 8C 11 D0 4C BC FE 00 FF
```

Képernyő-editor

Jól hasznosíthatják azok az alábbi programot, akik maguk szeretnék munka közben a képernyőt kezelni, mert az editorral készített képek felhasználhatók többek között a különböző játékok hátteréhez. A program futtatása után megadhatjuk, hogy kívánunk-e átalakított karaktereket. Ha igen, akkor azt, hogy melyik memóriaterületen vannak az átalakított karakterek. A K-változó értéke 1-7-ig terjedhet (350. sor). Tehát a K-változónak megfelelően 2048* K címen kezdődnek az átalakított karakterek. Megadjuk azt, hogy 1 képet vagy 3 képet editálunk. Ezután jelentkezik a kurzor. Bármit tetszés szerint írhatunk a képernyőre, hiszen működnek a kurzormozgató és a funkcióbillentyűk.

F1. Az aktuálisan kiírandó karakter színének változatása.

F2. Az előbbi karakter színének visszaállítása.

F5. Ha 3 képernyőt kezelünk, akkor a képernyőt (képet) változtatjuk. Nagyon érzékeny a gépi kódú megszakítás miatt ez a billentyű, ezért csak hirtelen megütéssel (gyors mozdulattal) kezelhetjük.

Hogyan történik a lemezre rögzítés?

A betöltés: Miután az editor (XØ) rögzitette a képeket, töltsük be. Az editor programból célszerű kiválasztani a gépi kódú program betöltésére szolgáló szubrutint. Ezt a részt használhatjuk saját BASIC programunkban is.

A gépi kódú rész használata futtatás után:

1. kép előhívása: SYS 49268: SYS 49448
2. kép előhívása: SYS 49328: SYS 49508
3. kép előhívása: SYS 49208: SYS 49388

Ha egy képet kirakunk a lemezre, majd behívjuk, akkor SYS 49328: SYS 49508 utasításokkal hívhatjuk elő.

DUNÁS V. ZSOLT

```
10 :REM ********************************
20 :REM *                              *
30 :REM *                              *
40 :REM *                              *
50 :REM *       KEPERNYO  EDITOR       *
60 :REM *                              *
70 :REM *                              *
80 :REM *  KESZITETTE : DUNAS V. ZSOLT *
90 :REM *                              *
100 REM *                              *
110 REM *          SZENTES,1987        *
120 REM *                              *
130 REM *                              *
140 REM *          COMMODORE 64        *
150 REM *                              *
160 REM *                              *
170 REM *                              *
180 REM ********************************
190 :
200 :
210 :
220 POKE56,60:POKE55,191
230 :
240 POKE53280,0:POKE53281,0:POKE650,128
250 IFPEEK(49152)<>120THENPRINT"■■■■■■■■■KEREM VARJON !":GOSUB960
260 REM **** ADATOK BEIRASA
270 :
280 PRINT"■■CHAR ATALAKITAS BETOLTVE ( I , N ) ? : ";:POKE204,0
290 GETA$:IFA$=""THEN 290
300 IFA$="I"THEN330
310 IFA$="N"THEN380
320 GOTO 290
330 POKE204,1:PRINT"■■ CHAR MEMORIA ( 1 - 7 ) ? : ";:POKE204,0
340 GETA$:IFA$=""THEN340
350 K=VAL(A$):IFK<1ORK>7THEN330
370 K=K-1:POKE53270,27:POKE53272,19+2*K
380 POKE204,1:PRINT"■■ OLDAL ( 1 VAGY 3 ) ? : ";:POKE204,0
390 GETA$:IFA$=""THEN390
400 IFA$="3"THEN PA=3:SYS 49152:GOTO 425
410 IFA$="1"THEN PA=1:GOTO 425
420 GOTO 390
425 IFPEEK(49152)=120THENFORY=1TO3:PRINT"■":POKE197,6:NEXTY
```

```
427 :
430 SI=0:POKE 53280,6:POKE 53281,6:RV=0
440 A=1024:K=86:S=55296:POKE204,1:PRINT"■⌐"
450 GOSUB 780
460 :
470 REM **** A KEPERNYOK KEZELESE
480 :
490 GETA$:IF A$=""THEN490
500 POKE A,T:K=86:POKES,TO
510 IFA$=CHR$(20)THENA=A-1:S=S-1:POKEA,32:GOTO770
520 IFA$="■"THENA=A+40:S=S+40:GOTO770
530 IFA$="⌐"THENA=A-40:S=S-40:GOTO770
540 IFA$="■"THENA=A+1:S=S+1:GOTO770
550 IFA$="■"THENA=A-1:S=S-1:GOTO770
560 IFA$="■"THENRV=1:GOTO770
570 IFA$="■"THENRV=0:GOTO770
580 IFA$=CHR$(13)THEN770
590 IFA$="■"THENPOKES,S1:S1=S1+1:IFS1>16THENS1=0
600 IFA$="■"THENPOKES,S1:S1=S1-1:IFS1<0THENS1=16
610 IFA$="■"ORA$="■"ORA$="■"ORA$="■"ORA$="■"ORA$="■" THEN770
620 IFA$="⌐"THENPRINT"⌐":GOTO 770
630 IFA$="■"THENPOKE53282,S1:GOTO770
640 IFA$="■"THEN 830
650 IFA$_"■"THENPOKE53283,S1:GOTO770
660 :
670 REM ****,ATSZAMITAS KEP - CHR KOD
680 :
690 K=ASC(A$)
700 IFK>=32ANDK=<63THENK=K
710 IFK>=64ANDK=<95THENK=K-64
720 IFK>=96ANDK=<127THENK=K-32
730 IFK>=160ANDK=<191THENK=K-64
740 IFK>128THENK=K-128
750 IFRV=1THENK=K+128
760 GOSUB780:A=A+1:S=S+1:K=86
770 GOTO 450
780 IFA<1024THENA=A+40:S=S+40
790 IFA>2023THENA=A-40:S=S-40
800 T=PEEK(A):POKE A,K
810 TO=PEEK(S):POKE S,S1:RETURN
820 :
830 REM **** NORMAL CHAR VISSZA ALLITAS
840 :
850 POKE A,T
860 IFPA=1THEN GOSUB910
870 POKE53280,245:POKE53281,246:POKE53272,21
880 POKE197,6:FORY=1TO1000:NEXTY:PRINT"⌐■■          ■  SAVE  KEPERNYO  "
890 GOTO1280
900 :
910 Q=0:FORI=22320TO23320:A=PEEK(1024+Q):B=PEEK(55296+Q)
920 POKEI,A:POKEI+1000,B:Q=Q+1:NEXTI
930 RETURN
940 :
950 :
960 REM **** GEPIKODU ADATOK
970 Q=0
980 READY:W=W+Y-128:IFY=-1THENIFW<>2791THENPRINT"⌐■■■■■  HIBAS ADATOK  ":END
990 IFY=-1THEN RETURN
1000 POKE49152+Q,Y:Q=Q+1:GOTO980
```

```
1010 :
1020 DATA120,169,13,141,20,3,169,192,141,21,3,88,96,166,197,224,6,240,3,76,49
1030 DATA234,76,130,193,234,160,0,185,0,4,153,192,63,185,255,4,153,191,64,185
1040 DATA254,5,153,190,65,185,253,6,153,189,66,136,208,229,96,160,0,185,192
1050 DATA63,153,0,4,185,191,64,153,255,4,185,190,65,153,254,5,185,189,66,153
1060 DATA253,6,136,208,229,96,160,0,185,0,4,153,168,67,185,255,4,153,167,68
1070 DATA185,254,5,153,166,69,185,253,6,153,165,70,136,208,229,96,160,0,185
1080 DATA168,67,153,0,4,185,167,68,153,255,4,185,166,69,153,254,5,185,165,70
1090 DATA153,253,6,136,208,229,96,160,0,185,0,4,153,144,71,185,255,4,153,143
1100 DATA72,185,254,5,153,142,73,185,253,6,153,141,74,136,208,229,96,160,0,185
1110 DATA144,71,153,0,4,185,143,72,153,255,4,185,142,73,153,254,5,185,141,74
1120 DATA153,253,6,136,208,229,96,160,0,185,0,216,153,120,75,185,255,216,153
1130 DATA119,76,185,254,217,153,118,77,185,253,218,153,117,78,136,208,229,96
1140 DATA160,0,185,120,75,153,0,216,185,119,76,153,255,216,185,118,77,153,254
1150 DATA217,185,117,78,153,253,218,136,208,229,96,160,0,185,0,216,153,96,79
1160 DATA185,255,216,153,95,80,185,254,217,153,94,81,185,253,218,153,93,82,136
1170 DATA208,229,96,160,0,185,96,79,153,0,216,185,95,80,153,255,216,185,94,81
1180 DATA153,254,217,185,93,82,153,253,218,136,208,229,96,160,0,185,0,216,153
1190 DATA72,83,185,255,216,153,71,84,185,254,217,153,70,85,185,253,218,153,69
1200 DATA86,136,208,229,96,160,0,185,72,83,153,0,216,185,71,84,153,255,216,185
1210 DATA70,85,153,254,217,185,69,86,153,253,218,136,208,229,96,238,60,3,174
1220 DATA60,3,224,3,208,5,169,0,141,60,3,234,234,234,234,174,60,3,224,0,208
1230 DATA12,32,25,192,32,206,192,32,116,192,32,40,193,174,60,3,224,1,208,12
1240 DATA32,86,192,32,10,193,32,176,192,32,100,193,174,60,3,224,2,208,12,32
1250 DATA146,192,32,70,193,32,56,192,32,236,192,234,234,234,169,1,133,204,76
1260 DATA49,234,-1
1270 :
1280 REM **** LEMEZRE IRAS
1290 :
1300 POKE828,0:IFPA=3THENPOKE828,1
1310 PRINT"■■■■■■■■■A PROGRAM NEVE :";:GOSUB1390
1315 Z=22320:X=24320:K=16320:IFPEEK(828)=1THENZ=16320:X=22320:K=16320
1320 OPEN 1,8,2,IN$+",PRG,WRITE"
1330 A2=INT(K/256):A1=K-(A2*256)
1340 PRINT#1,CHR$(A1)CHR$(A2);
1350 FORQ=ZTOX:D=PEEK(Q):PRINT#1,CHR$(D);:IFST<>0THENPRINT"■■■DISK HIBA":END
1360 NEXTQ
1370 CLOSE1:PRINT"■■ OK.":END
1380 REM ***** INPUT RUTIN ***
1390 X=19:Y=6
1400 IN$="":UN$="■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■":UN$=LEFT$(UN$,Y+1):RE=0
1410 GOSUB1450:IN$=IN$+A$:IFSW=20THENR=LEN(IN$):IFR=1THENIN$=LEFT$(IN$,R-1):SW=0
1420 IFSW=20THENIN$=LEFT$(IN$,R-2)
1430 IFRE=1THENR=LEN(IN$):IN$=LEFT$(IN$,R-1):RETURN
1440 GOTO 1410
1450 C=(C+1)AND1:PRINTUN$TAB(X)IN$CHR$(18+(128*C))" ":GETA$:IFA$=""THEN1450
1460 PRINTUN$TAB(X)IN$" "
1470 SW=ASC(A$):IFSW=13THENRE=1:GOTO1490
1480 IF SW<20 OR SW>95 THEN1450
1490 RETURN
```

Trükk Spectrumra

Régi téma az adatvédelem, de talán az alábbiak nem lesznek unalmasak a spectrumosoknak.

A biztonságos védelemhez feltétlenül szükséges az autostartos felvétel. Ezt a SAVE "NÉV" LINE sorszám utasítása nyújtja. A Merge "NÉV" beírására azonban az autostart hatástalan lesz, így ezt ki kell iktatni. Az alábbi két POKE nyomán a gép mintegy "befagy" a Merge név írására.

POKE(PEEK 23641+256 PEEK23642), PEEK 23635

POKE(PEEK 23641+256 PEEK23642)+1, PEEK 23636

Ezt az első tényleges programsorba célszerű beírni. A következő lépés a programozási fogások levédése. Az első programsort, melybe a "gyártó" nevét írjuk, a POKE 23756,Ø-val nullázhatjuk le. Ha még a Rem-et is ki akarjuk szedni, akkor ez a POKE 23759,32 utasítással történik.

A listázás letiltása: POKE 23755,1ØØ

engedélyezése: POKE 23755,Ø

A POKE 23755,1—32-ig parancs után csak az első sor "él", és az első sor sorszáma változik meg. A POKE 23757,1 után a gép editáláskor fagy be.

Ékezetes betűvel a kimentés: SAVE "VE"+CHR$8+CHR$21+CHR$1+"GE"

NEMESKÉRI ZSOLT

BASIC és gépi kód

Legutóbb a veremkezelő utasításokról volt szó. Ezzel az utasításcsoporttal befejeztem a 6502-es mikroprocesszor utasításainak bemutatását. Az ún. ismeretlen kódokkal több cikkben is találkozhatunk, ezért egyelőre nem szándékozom foglalkozni velük.

Most két olvasói észrevételről írok.

Még egyszer a LOAD-ról

Az előzmények: körülbelül egy évvel ezelőtt beszámoltam arról a „felfedezésemről", hogy egy gépi kódú program a LOAD parancs hatására a helyére töltődött, pedig nem adtam meg a másodlagos címet (µM 1986/11—12. szám). Pásthy Béla tatabányai olvasó levelében azt írta, hogy nála ez nem így van, és érdeklődött az eltérés oka felől. Válaszom az ez évi 6. számban jelent meg.

Újabb levelet kaptam Pásthy Bélától: tanácsom nem segített, mert bár kezdettől fogva a cikkemben ismertetett SAVE rutint használta, a LOAD utasítás nem tölti a programot az eredeti címre. Részletesen elemezve a gépben lejátszódó folyamatot bizonyította be, hogy az általam javasolt módszer miért nem vezet a kívánt eredményre. Megírta a megoldás módját is:Ø2DA-ra LDA #$Ø3-at célszerű írni.

Igaza van! Ezúton kérem elnézését a figyelmetlenségem miatti többletmunkáért. A probléma megoldásának legegyszerűbb módja — a fentiek szellemében — a következő. A SAVE rutin betöltése után POKE 731,3 utasítással megváltoztatjuk a másodlagos címet, majd a rutin leírásánál ismertetett módon (µM 1986/10. szám) a rutint szalagra visszük.

Most lássuk tévedésem hátterét. Amikor a SAVE rutint közlésre előkészítettem, nem gondoltam a LOAD utasítás különleges alkalmazására, mert nem is tudtam róla.

Egy alkalommal olyan C64-es gépen dolgoztam, amelyhez nem volt megbízható mágneslemezegység, ezért a programokat szalagon tároltam. Többek között azt a monitor/disassembler programot is, amellyel a jelen cikksorozathoz a disassembly listákat szoktam kinyomtatni. Ennek a programnak a betöltésekor történt az a bizonyos „felfedezésem", melyre fentebb utaltam. Csak egy dologról feledkeztem el: arról, hogy a mentést nem a SAVE rutinnal, hanem magával a monitorprogrammal végeztem el.

Kiszámított GOTO — rövidebben

Korábban említettem, hogy Székely Béla kollégám figyelmeztetett arra, hogy a kiszámított GOTO rutinom C16-os változata kétszer olyan hosszú, mint szükséges (µM 1987/2. szám). Megígértem, hogy visszatérek a témára.

A disassemblált rutint mutatja az *1. lista*, a *2. listán* pedig a C16 ROM-jának egy része látható. Nem nehéz észrevenni, hogy a kettő összevonható, és az eredeti rutin a *3. listán* bemutatott módon a felére rövidíthető. Ez a lehetőség csak C16-on van meg, és azért nem vettem észre, mert a másik két géptípuson használt algoritmust gépiesen alkalmaztam a C16-ra.

A ROM-ban tovább bogarászva fedeztem fel a *4. listán* lévő részletet. Megnyugodtam; nemcsak én hagytam ki a rövidítési lehetőségeket, hanem az interpreter készítői is. A két utasítás helyett elég lenne egy is: JSR $9DE1.

BARNA LÁSZLÓ

```
033c   209194   jsr $9491
033f   201493   jsr $9314
0342   20e49d   jsr $9de4
0345   4c508d   jmp $8d50
```

1. lista

2. lista

```
...
9dde   209194   jsr $9491
9de1   201493   jsr $9314
9de4   a566     lda $66
...
```

```
033c   20de9d   jsr $9dde
033f   4c508d   jmp $8d50
```

3. lista

4. lista

```
...
9dd2   201493   jsr $9314
9dd5   20e49d   jsr $9de4
...
```

Először is elnézést kérek az olvasóktól: a sorozat 1. részében, amely az 1987/7. számban jelent meg, egy programhiba van; eddig senki nem vette észre (én sem), a program viszont így nem működik. A hibás sor helyesen:

3Ø JP LDIRQ2

Remélem, több ilyen hiba nem lesz, de ha mégis, kérem az olvasót, írja meg.

A manószerkesztő program további rutinjait mutatom be ebben a részben (lásd *1. lista*). Ezek még mindig alaprutinok. Programmá való összefűzésükre a sorozat egy későbbi részében kerül sor.

A program interaktív kapcsolatot tart a felhasználóval. Ennek megvalósításához sokféle I/O rutinra van szükség. Közülük csak néhány készült kimondottan a manószerkesztőhöz, többségükben univerzálisak. Most ezeket mutatom be részletesen.

Képernyőre író rutinok

OPEN2. Megnyitja a képernyőcsatornát. A ROM-ot hívja. Minden regiszter tartalmát elrontja. A scroll-számlálót 255-re állítja, ami azt jelenti, hogy 255 alkalommal kérdezés (scroll?) nélkül sort emel. Ha kiíratás közben időnként meghívjuk az OPEN2 rutint, nem lesz „scroll?". Megjegyzés: ha le van tiltva a megszakítás (DI), akkor életveszélyes másképp kiírani, mert a „scroll?" kérdésre befagy. Ennek az az oka, hogy Spectrumon a megszakításrutin végzi a billentyűzet ciklikus leolvasását.

Primón nincs szükség mindezekre a trükkökre.

EMIT. A karakterkiviteli rutin. A karakterkód az A regiszterben van. A Spectrum ROM-ot hívja, minden regisztert megőriz, csak az AF-et rontja el. A Primóra átírt változat ugyanezt teszi.

ATBC. A B-edik sor C-edik oszlopára állítja a kurzort. Ha B nagyobb 21-nél, vagy C nagyobb 31-nél, akkor hibaüzenetet kapunk.

WRITE. Kiír egy szöveget, amelynek a kezdőcíme a HL regiszterben van, a végét pedig EOL (=80) karakter jelzi. A szöveget a megnyitott csatornára viszi ki, ez Spectrumon lehet a képernyő alsó része vagy felső része, Primón csak egyszerűen a képernyő.

MSG. Kiírja azt a szöveget, amelyet a CALL MSG utasítás után DEFB-vel leraktunk. A szöveg végét itt is EOL jelzi. Ha azt akarjuk, hogy programunk elszálljon, akkor felejtsük le az EOL-t a szöveg végéről.

DECIHL. Kiírja decimálisan a HL regiszterpár tartalmát. Az E regiszterben legyen az a karakter, amelyet a szám elején álló nullák helyett kell nyomtatni, például

Z80 programok haladóknak Spectrumra és Primóra

6. I/O és egyéb rutinok

```
;-----------------------------------;
;                                   ;
;     Sprdef 2.1   I/O rutinok      ;
;       UHI-software BME            ;
;                                   ;
;-----------------------------------;
SCR     EQU  $4000      ;kepernyocim
SCBUF   EQU  $D000      ;kepbuffercim
BUFEND  EQU  $E7FF      ;puffer vege
SCRLEN  EQU  $1800      ;kepmeret
SCREND  EQU  $57FF      ;kepernyo veg
AADD    EQU  $484A      ;A ablak cime
AHADD   EQU  $8A        ;
BADD    EQU  $4010      ;B ablak cime
A00     EQU  $4050      ;A koord. y,x
B00     EQU  $3380      ;B koordinata
O00     EQU  $0F00      ;0 koordinata
INK     EQU  16         ;vezerlo chr
PAP     EQU  17         ;"
AT      EQU  22         ;"
CR      EQU  13         ;carriage ret
EOL     EQU  $80        ;end of line
PIXADD  EQU  $22AC      ;spectrum rom
EMIT    EQU  $0010      ;"
UDG     EQU  $5C7B      ;spectrum uds
BSP     EQU  $8F00      ;
SIZ     EQU  32         ;manomagassag
TXT     DEFS 2          ;
SPRADD  DEFS 2          ;manoalakfile
FILEN   DEFS 2          ; " hossza
SP1     DEFS 1          ;listazando
SPE     DEFS 1          ;utolso mano
SPOS    DEFS 2          ;pont-cursor2
POS     DEFS 2          ;pont-cursor1
CDS     DEFS 2          ;
INBUF   DEFS 6          ;szovespuffer
SBUFF   DEFS 128        ;mano-puffer
PATTS   DEFS 64         ;mintazatok
;-----------------------------------;
OPEN2   LD   A,2        ;kepcsatorna
        CALL $1601      ;open-chan
        LD   (IY+82),-1 ;scrolls
        RET             ;
;-----------------------------------;
ATBC    LD   A,AT       ;vezerles
        CALL EMIT       ;chr-out
        LD   A,B        ;sor
        CALL EMIT       ;
        LD   A,C        ;oszlop
        CALL EMIT       ;
        RET             ;
;-----------------------------------;
MSG     EX   (SP),HL    ;cim HL-be
        CALL WRITE      ;kiiras
        EX   (SP),HL    ;vegcim mege
        RET             ; visszater
;-----------------------------------;
WRITE   PUSH AF
WRITE1  LD   A,(HL)     ;kov chr
        INC  HL         ;cim noveles
        CP   EOL        ;vege?
        JR   Z,WRITEE   ; ha igen
        CALL EMIT       ;kiirja
        JR   WRITE1     ;folytatas
WRITEE  POP  AF
        RET
;-----------------------------------;
DECIHL  PUSH BC         ;
        PUSH AF         ;
        PUSH HL         ;
        LD   BC,-10000  ;elso jegy
        CALL OUTNO      ;   kiirja
        LD   BC,-1000   ;masodik jegy
        CALL OUTNO      ;   kiirja
DECI3   LD   BC,-100    ;harmadik
        CALL OUTNO      ;   kiirja
        LD   BC,-10     ;negyedik
        CALL OUTNO      ;   kiirja
        LD   A,L        ;5.,utolso
        CALL OUTNO3     ;0-t is kiir
        POP  HL
        POP  AF
        POP  BC
        RET
OUTNO   XOR  A          ;szamlalo 0
OUTNO1  ADD  HL,BC      ;"kivonas"
        INC  A          ;szamlalo nov
        JR   C,OUTNO1   ;HL)0, tovabb
        SBC  HL,BC      ;"restaural"
        DEC  A          ;szamlalo jav
        JR   Z,OUTNO2   ;0 eseten
OUTNO3  LD   E,$30      ;mar kell "0"
OUTNO2  ADD  A,E        ;ascii
        CALL EMIT       ;kiir
        RET
```

```
;-----------------------------------;
DECIA   LD    E," "       ;elejen space
        PUSH  BC
        PUSH  AF
        PUSH  HL
        LD    H,0         ;HL=A
        LD    L,A
        JR    DECI3       ;kiir 3 szje
;-----------------------------------;
SPACE   PUSH  AF
        LD    A," "
        CALL  EMIT        ;kiir
        POP   AF
        RET
;-----------------------------------;
PLOTBC  PUSH  BC
        PUSH  HL
        PUSH  AF
        LD    A,#AF       ;y hatar
        CALL  PIXADD      ;kepelem cime
        LD    B,A         ;eltolasok sz
        LD    A,1         ;pont-maszk
        INC   B           ;
PLBC1   RRCA              ;maszk tolo
        DJNZ  PLBC1       ;   ciklus
        XOR   (HL)        ;over 1 mod
        LD    (HL),A      ;kepre tesz
        POP   AF
        POP   HL
        POP   BC
        RET
;-----------------------------------;
CLS     PUSH  BC
        LD    B,#18       ;sorok szama
        CALL  #0E44       ;cl-line rom
        POP   BC
        RET
;-----------------------------------;
SPRITE  PUSH  BC
        PUSH  DE
        PUSH  HL
        PUSH  IX
        PUSH  DE          ;IX=DE
        POP   IX          ;...
        LD    A,#9F       ;y hatar
        CALL  PIXADD      ;kepelem cime
        DI                ;spectrumon
        ;IY hasznalata eseten kell DI
        ;mert az INT rutin is hasznal
        PUSH  IY          ;
        PUSH  HL          ;IY=HL
        POP   IY          ;...
        LD    B,SIZ       ;manomagassag
        LD    C,A         ;jobbra tolas
SPR1    PUSH  BC          ;
        LD    D,(IX+0)    ;DEHL-be ol-
        LD    E,(IX+1)    ; vassa a ma-
        LD    H,(IX+2)    ; no egy so-
        LD    L,(IX+3)    ; rat...
        LD    A,C         ;tolasok sz
        OR    A           ;vizsgalat *
        LD    A,0         ;A=0
        JR    Z,SPR2      ;*  ha 0 volt
        LD    B,C         ;tolasok sz
SPRSH   SRL   D           ;tolo ciklus
        RR    E           ; a mano egy
        RR    H           ; sorat egy-
        RR    L           ; szerre tol-
        RRA               ; ja jobbra.
        DJNZ  SPRSH       ;ciklus vege
SPR2    XOR   (IY+4)      ;kepernyore
        LD    (IY+4),A    ; irja az e-
        LD    A,L         ; gesz sort
        XOR   (IY+3)      ; over 1 mod-
        LD    (IY+3),A    ; ban (xor)
        LD    A,H         ; ...
        XOR   (IY+2)      ; ...
        LD    (IY+2),A    ; ...
        LD    A,E         ; ...
        XOR   (IY+1)      ; ...
        LD    (IY+1),A    ; ...
        LD    A,D         ; ...
        XOR   (IY+0)      ; ...
        LD    (IY+0),A    ; ...
        CALL  IYDOWN      ;kepcim ujsor
        LD    BC,4        ;manocim ujs
        ADD   IX,BC       ;...
        POP   BC          ;ciklusvalt
        DJNZ  SPR1        ;amig van sor
        POP   IY          ;IY a regi,
        EI                ; tehat EI
        POP   IX          ;
        POP   HL          ;
        POP   DE          ;
        POP   BC          ;
        RET               ;
;-----------------------------------;
IYDOWN  DEFB  #FD         ;IY elokod
        INC   H           ;INC YH
        DEFB  #FD         ;
        LD    A,H         ;LD A,YH
        AND   #07         ; a rutin mu-
        RET   NZ          ; kodese e-
        PUSH  BC          ; gyebkent
        LD    BC,#F820    ; megegyezik
        ADD   IY,BC       ; a HLDOWN
        POP   BC          ; mukodesevel
        DEFB  #FD
        LD    A,H
        AND   #07
        RET   Z
        LD    A,#07
        DEFB  #FD
        ADD   A,H
        DEFB  #FD
        LD    H,A
        RET
;-----------------------------------;
PAUSE0  PUSH  BC
        RES   3,(IY+48)   ;capslock ki
        RES   5,(IY+1)    ;nem volt bil
        EI                ;biztonsag
        LD    BC,0        ;parameter 0
        CALL  #1F3D       ;rom PAUSE BC
        LD    A,(#5C08)   ;last key
        POP   BC
        RET
```

```
;-----------------------------------;
WAIT    EI              ;biztonsag
        PUSH BC
        RES  5,(IY+1)   ;nem volt b
        LD   BC,15      ;parameter 15
        CALL #1F3D      ;PAUSE BC
        POP  BC
        RET
;-----------------------------------;
GET     CALL MSG        ;kiir egy
        DEFB 18,1,"_",18,0   ;villogo
        DEFB 8,EOL      ;cursort
        CALL PAUSE0     ;bill var
        CALL PIP        ;csipog
        PUSH AF
        CALL MSG        ;Ezzel torli
        DEFB " ",8,EOL  ; a cursort.
        POP  AF
        RET
;-----------------------------------;
GETNO   CALL GET        ;bill var,cur
        CALL CLRIN      ;kerdes torl
        SUB  "0"        ;ascii==)szam
        RET  C          ;hiba ha < 0
        CP   8          ;felso hatar
        CCF             ;cf ha > 7
        LD   L,A        ;HL=A szam
        LD   H,0        ;...
        RET
;-----------------------------------;
OPENIN  PUSH DE
        PUSH BC
        PUSH HL
        XOR  A          ;csatornaszam
        CALL #1601      ;open-chan
        CALL MSG        ;pozicional
        DEFB AT,1,0,EOL ;...
        POP  HL
        POP  BC
        POP  DE
        RET
;-----------------------------------;
CLRIN   PUSH BC
        PUSH AF
        LD   B,2        ;sorok szama
        CALL #0E44      ;cl-line rom
        POP  AF
        POP  BC
        RET
;-----------------------------------;
TEXTIN  PUSH HL
        PUSH BC
        LD   HL,INBUF   ;szovegpuffer
        LD   B,C        ;max hossz
TIN1    CALL GET        ;bill olvasas
        CP   CR         ;enter?
        JR   Z,TINE     ; igen
        CP   " "        ;space?
        JR   C,TIN8     ; kisebb, bs
        CP   #80        ;max chr
        JR   NC,TIN1    ; nem ascii
        INC  B          ;B vizsgalat
        DEC  B          ;...
        JR   Z,TIN1     ;ha zerus
        LD   (HL),A     ;pufferbe ir
        CALL EMIT       ;kepre is ir
        INC  HL         ;puffercim
        DEC  B          ;chr szamlalo
        JR   TIN1       ;folytatas
TIN8    LD   A,B        ;chr szaml
        CP   C          ;sor eleje?
        JR   Z,TIN1     ; igen
        LD   A,8        ;backspace
        CALL EMIT       ;visszaler
        INC  B          ;szamlalo
        DEC  HL         ;puffercim
        JR   TIN1
TINE    LD   (HL),A     ;cr beiras
        POP  BC
        POP  HL
        CALL CLRIN      ;input torles
        RET
;-----------------------------------;
VAL     PUSH BC
        PUSH DE
        LD   DE,INBUF   ;szovegpuffer
        LD   HL,0       ;eredmeny 0
VAL1    LD   A,(DE)     ;kov char
        INC  DE         ;pointer nov
        CP   CR         ;sorvege?
        JR   Z,VALX     ; igen
        CP   " "        ;space?
        JR   Z,VAL1     ;elejet keres
VALD    CALL HL10       ;HL=HL*10
        JR   C,VALX     ; tulcsordul
        CP   #3A        ;"9"+1
        JR   NC,VALX    ; )=, hiba
        SUB  "0"        ;ascii==)szam
        JR   C,VALX     ; (0, hiba
        LD   B,0        ;BC=A
        LD   C,A        ;...
        ADD  HL,BC      ;HL=HL+BC
        JR   C,VALX     ; tulcsordul
        LD   A,(DE)     ;kov chr
        INC  DE         ;pointer nov
        CP   CR         ;sorvege?
        JR   Z,VALE     ; vege,ok
VAL2    JR   VALD       ;folytat
VALX    SCF             ;hiba: cf
VALE    POP  DE         ;vege
        POP  BC
        RET
;-----------------------------------;
HL10    ADD  HL,HL      ;HL=HL*2    2
        LD   B,H        ;BC=HL      2
        LD   C,L        ;...
        ADD  HL,HL      ;HL=HL*2    4
        ADD  HL,HL      ;HL=HL*2    8
        ADD  HL,BC      ;HL=HL+BC  10
        RET
;-----------------------------------;
INP     CALL OPENIN     ;also resz
        CALL CLRIN      ;torles
        POP  HL         ;szoveg cime
        CALL WRITE      ;kiirja
        PUSH HL         ;visszateresi
```

```
        CALL TEXTIN     ;szoveg olv
        CALL VAL        ;kiertekeles
        RET
;-------------------------------;
PIP     PUSH AF
        PUSH BC
        PUSH HL
        PUSH IX
        LD   DE,10      ;hang hossz
        LD   HL,200     ;magassag
PIP1    CALL #03B5      ;ROM beeper
        POP  IX
        POP  HL
        POP  BC
        POP  AF
        RET
;-------------------------------;
CURX    LD   BC,(POS)   ;plot-cur pos
        CALL BCOORD     ;coord konver
        LD   A,#0C      ;
        CALL CURX_1     ;
        LD   A,#05      ;
        CALL CURX_1     ;
        LD   A,#0D      ;
        CALL CURX_1     ;
        LD   A,#04      ;
        CALL CURX_1     ;
        RET
CURX_1  LD   (CURX2),A  ;onmodositas!
        LD   A,3        ;3-szor
CURX11  CALL PLOTBC     ;kitesz
CURX2   NOP             ;ler
        DEC  A          ;ciklus-
        JR   NZ,CURX11  ; zaras
        RET
;-------------------------------;
CURY    LD   BC,(SPOS)  ;Y cursor pos
        CALL BCOORD     ;coord conv
        INC  C          ;a mezo koze-
        DEC  B          ; pere kitesz
        CALL PLOTBC     ; egy 4*4-es
        INC  C          ; pontot
        CALL PLOTBC     ;...
        DEC  B          ;...
        CALL PLOTBC     ;...
        DEC  C          ;...
        CALL PLOTBC     ;...
        RET
;-------------------------------;
KEYIN   CALL CURX       ;X cursor ki
        CALL CURY       ;Y cursor ki
        CALL WAIT       ;var 320ms-ot
        PUSH AF         ;bill-kod
        CALL CURX       ;X cursor le
        CALL CURY       ;Y cursor le
        POP  AF         ;bill-kod
        JR   NZ,KEY2    ;volt bill
        CALL WAIT       ;var 320ms-ot
        JR   Z,KEYIN    ;nincs billny
KEY2    RES  5,(IY+1)   ;billenty-bit
        LD   A,(#5C08)  ;last key
        CP   CR         ;enter?
        RET  NZ         ;nem
        LD   A,"E"      ;enter==)"E"
        RET
;-------------------------------;
ACOORD  PUSH HL
        LD   HL,A00     ;"A" mezo y,x
        ADD  HL,BC      ;relativ=)abs
        LD   B,H        ;BC=HL
        LD   C,L        ;...
        POP  HL
        RET
;-------------------------------;
BCOORD  SLA  B          ;B=B*4
        SLA  B          ;...
        SLA  C          ;C=C*4
        SLA  C          ;...
        PUSH HL
        LD   HL,B00     ;"B" mezo y,x
        ADD  HL,BC      ;relativ=)abs
        LD   B,H        ;BC=HL
        LD   C,L        ;...
        POP  HL
        RET
;-------------------------------;
LISTSP  PUSH HL
        PUSH AF
        CALL OPEN2      ;kepernyo
        CALL MSG        ;kiirja:
        DEFB AT,17,0,PAP,8
        DEFB 6,6,6,6,6,6
        DEFB 6,6,6,6,6
        DEFB AT,15,0
        DEFM " Old spri"
        DEFM "tes:   "
        DEFB AT,17,0,EOL
        LD   A,(SP1)    ;listazando
        LD   B,8        ;innen 8-at
LIST1   CALL DECIA      ;mano sorszam
        INC  A          ;kov
        CALL SPACE
        DJNZ LIST1      ;sorszamciklu
        LD   A,(SP1)    ;listazando
        CALL OLDADD     ;manocimet ad
        EX   DE,HL      ;DE=manocim
        LD   BC,000     ;"0" mezo y,x
LIST2   CALL SPRITE     ;kirakja
        LD   A,C        ;x=x+meret
        ADD  A,SIZ      ;manomeret
        JR   Z,LISTE    ;megvan a 8
        LD   C,A        ;x koordinata
        EX   DE,HL      ;HL=manocim
        LD   DE,#0080   ;mano bajtban
        ADD  HL,DE      ;kov manocim
        EX   DE,HL      ;DE=manocim
        JR   LIST2
LISTE   POP  AF
        POP  HL
        RET
;-------------------------------;
LISTPT  CALL OPEN2
        CALL MSG
        DEFB AT,15,1,PAP,8
        DEFM " Patterns:"   ;a lista
        DEFM "      "       ; cime
        DEFB CR,CR
        DEFM "  0   1   2" ;minta-
```

```
477             DEFM "   3   4   "   ;sorsza-
478             DEFM "5   6   7 "   ;mok
479             DBFB BDLPATTS       ;mintak cime
481             LD   (UDG),HL       ;spectrum UDG
482             LD   C,4            ;4 sorban
483 LSPT1       LD   A,144          ;UDG: 144...
484 LSPT2       LD   B,4            ;mindet 4szer
485 LSPT3       PUSH AF             ;EMIT rontja
486             CALL EMIT           ;kiirja
487             POP  AF
488             DJNZ LSPT3
489             INC  A              ;kov minta
490             CP   152            ;utolso utani
491             JR   NZ,LSPT2       ;meg nem
492             DEC  C              ;sorciklus
493             JR   NZ,LSPT1       ;...
494             RET
495 ;-----------------------------------;
496 CLEAR       CALL SAVESP         ;mano mentes
497 CLEAR1      CALL MSG            ;kiirja:
498             DEFB PAP,7,INK,0    ;feher-fekete
499             DEFB 19,1,EOL       ;bright 1
500             LD   BC,$0A0A       ; 10,10
501 CLEAR2      CALL ATBC           ;pozicional
502             CALL MSG
503             DEFM "     "
504             DEFB EOL
505             INC  B              ;kov. sor
506             LD   A,B            ;
507             CP   14             ;utolso utani
508             JR   NZ,CLEAR2      ;meg nem
509             LD   BC,$0010       ;  0,10
510 CLEAR3      CALL ATBC           ;pozicional
511             LD   A,6            ;16 space
512             CALL EMIT           ;kiirja
513             INC  B              ;kov. sor
514             LD   A,B            ;
515             CP   16             ;utolso utani
516             JR   NZ,CLEAR3      ;meg nem
517             CALL MSG            ;kiirja:
518             DEFB 19,0,EOL       ;bright 0
519             RET
520 ;-----------------------------------;
521 DRAW        LD   A,C            ;Ez a rutin
522             CP   B              ;a spectrum
523             JR   NC,DRAWX       ;ROMjaban le-
524             LD   L,C            ;vo vonalhuzo
525             PUSH DE             ;rutin gyor-
526             XOR  A              ;sitott val-
527             LD   E,A            ;tozata...
528             JR   DRAWL
529 DRAWX       OR   C
530             RET  Z
531             LD   L,B
532             LD   B,C
533             PUSH DE
534 DRAWL       LD   H,B
535             LD   A,B
536             RRA
537 DRAW1       ADD  A,L
538             JR   C,DRAWD
539             CP   H
540             JR   C,DRAWH
541 DRAWD       SUB  H
542             LD   C,A
543             EXX
544             POP  BC
545             PUSH BC
546             JR   DRAWS
547 DRAWH       LD   C,A
548             PUSH DE
549             EXX
550             POP  BC
551 DRAWS       LD   HL,(CDS)
552             LD   A,B
553             ADD  A,H
554             LD   B,A
555             LD   A,C
556             INC  A
557             ADD  A,L
558             JR   C,DRAWR
559             JR   Z,DRAWE
560 DRAWP       DEC  A
561             LD   C,A
562             LD   (CDS),BC
563             CALL PLOTBC
564             EXX
565             LD   A,C
566             DJNZ DRAW1
567 DRAWE       POP  DE
568             RET
569 DRAWR       JR   Z,DRAWP
570             JR   DRAWE
571 ;-----------------------------------;
```

space, "Ø" vagy bármi. Ha ez egy —1 kódú (# FF) karakter, akkor semmit nem ír oda. Legyen például HL tartalma 22, E tartalma "Ø" vagyis 48. Ekkor a DECIHL rutin "ØØ22"-t ír ki. Ha ugyanebben a példában E tartalma —1 (255), akkor az eredmény "22". Ha E tartalma " " vagyis 32, akkor a képernyőre " 22" íródik ki. Ez táblázatok kitöltésénél nagyon hasznos lehet.

DECIA. Az akkumulátortartalmat írja ki decimálisan. Az E regiszterbe szóközt tölt, tehát a számot egy háromkarakteres mezőben jobbra illesztve írja ki.

SPACE. Kiír egy szóközt, közben minden regisztert megőriz.

PLOTBC. A B=y, C=x koordinátájú pontot fordítja ellenkezőjére (ki-, illetve bekapcsolja).

CLS. Törli a teljes képernyőt. Spectrumon ROM-rutint hív.

SPRITE. Ez egy primitív manórajzoló rutin; a BC koordinátákra kirajzolja a DE című, 32×32 pixeles manóalakot. A képre KIZÁRÓ VAGY művelettel keveri rá, tehát eltűntetni úgy lehet, hogy ismételten kirajzoltatjuk ugyanoda ugyanazt a manót. Ekkor az eredeti képet kapjuk vissza. Ez egy viszonylag gyors, egyszerűen kezelhető kis rutin, játékprogramokban ezt is sok sikerrel lehet alkalmazni. Használata viszont körülményesebb, mint a már bemutatott manókezelőé, és sokkal kevesebbet tud. Természetesen ehhez is jók a manóalakszerkesztővel összeállított manók, de maszkot ne rajzoljunk hozzájuk.

A billentyűzetről olvasó rutinok

PAUSEØ. Billentyűnyomásra vár. A lenyomott billentyű ASCII kódja az A regiszterbe kerül. A Spectrum ROM-ot használja.

WAIT. 320 ms-ot vár, de ha ezalatt billentyűnyomás történt, abbahagyja a várakozást; a billentyűkód az A regiszterbe kerül.

GET. Kiír egy villogó kurzort, billentyűnyomásra vár, majd a kurzort eltünteti. A megnyomott billentyű kódját most is az akkumulátorban kapjuk.

GETNO. Egy Ø és 7 közötti egész szám beolvasására alkalmas. Kiírja a kurzort, de bármely billentyű megnyomásakor el is tűnteti. Ha a billentyű nem a megadott tartományban van, carry flaggel jelzi a hibát. A szám értéke a HL regiszterpárban lesz.

OPENIN. Megnyitja kiírásra a képernyő alsó részét. Primón minden bűvészkedés nélkül a kurzort a 23. sorra kell állítani (új karakterkiviteli rutin!).

CLRIN. Törli az alsó részt, azaz az alsó két sort.

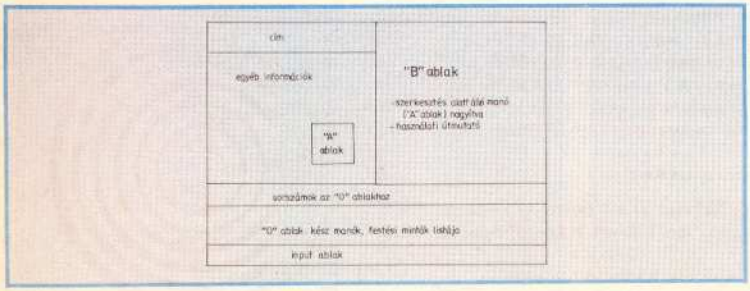
TEXTIN. Az INBUF nevű bemeneti pufferbe beolvas egy szöveget a billentyűzetről. A szöveg legnagyobb lehetséges hosszát a C regiszterbe töltsük. Ez ne legyen nagyobb, mint az INBUF hossza. A rutin szerkesztési szolgáltatása mindössze annyi, hogy bármely vezérlőgombra visszalépést csinál. Billentyűelfogadáskor csipog. A bevitel a képernyőn ott látszik, ahol a kurzor állt, tehát ha az alsó részen akarunk beolvasni, először az OPENIN rutint kell meghívni.

Egyéb rutinok

Ezek a manókezelő speciális rutinjai, de talán lehet belőlük ötleteket meríteni. A manószerkesztő a felhasználóval való kapcsolat céljából négy képernyőablakon jeleníti meg a különböző adatokat, ábrákat. Az „A" jelű ablakban (lásd az *ábrát*) az éppen szerkesztés alatt álló manó látható. A „B" ablakban ennek négyszeresre nagyított másolatát találjuk, de ide kerül a H (help) billentyű megnyomásakor a használati utasítás is. Az „0" ablakon a már elkészült manók láthatók (old sprites), vagyis a manóalakfájl egy részlete (8 db fér el). Ugyanebben az ablakban jelenik meg a választható minták listája is.

Ezekhez az ablakokhoz kapcsolódnak az alábbi rutinok.

CURX, CURY. Kétféle pontkurzort használ a manószerkesztő. Ezeket teszik ki vagy veszik le ezek a rutinok. A „B" ablakban jelennek meg, a POS vagy OLDPOS koordinátájú helyen (az alsó bájt az x, a felső bájt az y koordináta, az ablakon belül).



VAL. Kiszámolja az INBUF-ban levő szöveg számértékét. Ha decimális számként nem értelmezhető szöveget talál, eredményül az első hibás karakterig értelmezhető számot adja, és carryvel jelez. Az eredményt a HL-be teszi. Ezen belül a HL1Ø rutin a HL értékét szorozza tizzel. Hasonló módszerrel bármilyen kis egésszel való szorzást meg lehet valósítani.

INP. Megvalósít egy számbevitelt a képernyő alsó részén. A C regisztert a TEXTIN rutinnál leírtakhoz hasonlóan töltsük be. A beolvasott szám a HL-be kerül. Hiba esetén carryvel jelez. A beolvasás előtt kiír egy kommentárt, amit a CALL INP utasítás mögött helyeztünk el DEFB vagy DEFM pszeudoutasításokkal. A szövegvéget itt is EOL karakter jelzi. Használatára a nyolcadik részben láthatunk példát.

PIP. Hangjelzés, ami a billentyűleütést kíséri.

KEYIN. Kiteszi és villogtatja a két pontkurzort, billentyűnyomást vár. Akkor a pontkurzorok eltűnnek, a billentyűkód az akkumulátorba kerül.

ACOORD, BCOORD. Ha BC-ben y, x koordináták vannak, amelyek relatív koordináták az „A" vagy „B" ablakon belül, ezek a rutinok a tényleges képernyő-koordinátát számolják ki. Az eredményt is a BC-ben kapjuk.

LISTPT, LISTSP. Az „0" ablakba listázza a mintákat, illetve a manóalakfájl megfelelő részletét.

CLEAR. Törli az „A" és „B" ablakokat.

DRAW. Ez az egyenesrajzoló rutin, itt még semmihez nem kapcsolódik. Bemenő paraméterei bonyolultak, használatára a következő részben látunk példát.

A következő részben a Primóhoz való módosításokat mutatom be.

UHERKOVICH PÉTER

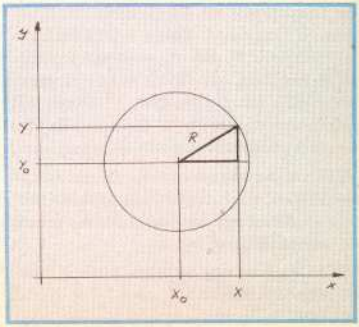
A számítógépes grafikában görbéket kétféleképpen rajzolhatunk a képernyőre. Az első lehetőség az, hogy az egyenesrajzolás eljárásához hasonlóan minden görbéhez egy-egy görbegeneráló eljárást írunk, amely lényegében azt teszi, hogy a kezdeti képpontból kiindulva meghatározza, hogy a soron következő képpontot hová kell helyezni, majd fel is rajzolja. Ennek az eljárásnak, ha az adott görbére egyáltalán sikerül találnunk megfelelő algoritmust, az az előnye, hogy nagyon szép, sima rajzolatot készít. Hátránya az, hogy minden egyes görbéhez új algoritmust kell kiagyalnunk, valamint lassúsága, hiszen minden képpontban ki kell számítanunk a görbét leíró, esetleg bonyolult függvény értékét.

A másik lehetőségünk az, hogy a görbe leírásához valamilyen alkalmas, paraméteres egyenletrendszert választunk, majd csak néhány, mindig az előzőhöz közel eső görbepontnak számítjuk ki a koordinátáit, és a kapott pontokat rövid, egyenes szakaszokkal kötjük össze. Az alkalmas egyenletrendszeren azt értjük, hogy az olyan legyen, hogy a paraméter egyenletes léptetése lehetőleg azonos vagy közel azonos távolságban lévő pontok sorozatát határozza meg.

Ez a módszer elvileg pontatlanabb az előzőnél, mivel a két pont között a görbe helyett mindig csak a húrját rajzolja fel. Ha azonban a pontok távolsága kicsi, a hiba rendszerint észrevehetetlen marad. Csak olyankor hibázik az eljárás jelentősebben, ha a görbe görbületi sugara a szomszédos kiszámított pontok távolságával összemérhető. Viszont sokkal gyorsabban dolgozik, mint az első módszer, és teljesen általánosan alkalmazható minden szabályos görbére.

A fentiek kissé elvontaknak tűnhetnek, ezért lássuk a kör példáját. Az első módszerre nem készítünk Pascal nyelvű eljárást; ha valaki kíváncsi egy ilyen algorit-

1. ábra



Számítógépes grafika Pascalban

KÖR

ZX–SPECTRUM

A Mikroszámítógép Magazin 87/11-es számában bemutattuk, hogyan lehet egyeneseket rajzolni a ZX—Spectrumon a Hisoft Pascal fordítóval. Most, felhasználva a már ismertetett eljárásokat, a kör, körív, ellipszis programozására mutatunk példát.

musra, figyelmébe ajánljuk a Spectrum ROM körgenerátorát, melynek részletes magyarázata például a „ZX—Spectrum ROM programja" című kiadványban megtalálható.

Ami a kör egyenletét illeti, itt több lehetőség merülhet fel. Kiindulhatunk a kör közismert egyenletéből, mely a Pitagorasz-tétel felhasználásával írható fel az *1. ábra* jelöléseivel:

$(X-X_0)^2+(Y-Y_0)^2=R^2$,

majd ebből Y-t kifejezve az

$Y=Y_0+\sqrt{R^2-(X-X_0)^2}$

egyenletet kapjuk. Ha a második módszert akarnánk erre az egyenletre alkalmazni, akkor X-et kellene léptetni. Az ilyen módon meghatározott pontok azonban nincsenek egymástól azonos távolságra: a vízszintes érintőjű pontok közelében vannak legközelebb, míg a függőleges érintőjű pontok környékén távol esnek egymástól. Ezért, ha ezt az egyenletet programoznánk, az ábrázolt görbét nem tartanánk körnek. (Ha az olvasónak kedve van hozzá, próbálja ki!)

Felírhatjuk a kör paraméteres egyenletrendszerét a *2. ábra* jelöléseivel (paraméter a φ középponti szög):

$X=X_0+R*\cos\varphi$ és

$Y=Y_0+R*\sin\varphi$.

Ez az egyenletrendszer már szép, szabályos kör képét fogja adni, ha az egyes pontok meghatározásához a φ paramétert egyenletes lépésközzel léptetjük. Csak egy baj lesz: a sok szögfüggvény számítása lelassítja a rajzolást.

Gyorsabb algoritmusra vezet az alábbi gondolatmenet (lásd *3. ábra*). Legyen a koordinátarendszerben egy O és egy P pontunk. A P pont koordinátáit az O pont koordinátáival, az OP szakasz R hosszával és φ hajlásszögével így fejezhetjük ki:

$X=X_0+R*\cos\varphi$,

$Y=Y_0+R*\sin\varphi$

Forgassuk el a P pontot O körül $\Delta\varphi$ szöggel P'-be. Ekkor a P koordinátáit így írhatjuk fel:

$X'=X_0+R*\cos(\varphi+\Delta\varphi)$ és

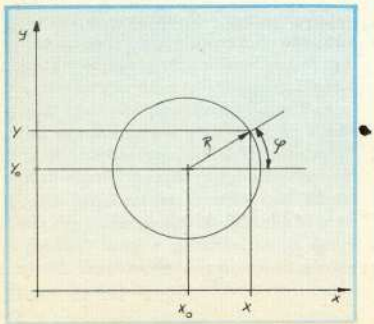
$Y'=Y_0+R*\sin(\varphi+\Delta\varphi)$.

A trigonometriából ismeretes, hogy két szög összegének szinusza és koszinusza így írható:

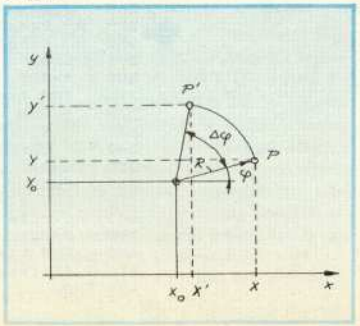
$\sin(\varphi+\Delta\varphi)=\sin\varphi*\cos\Delta\varphi+\cos\varphi*\sin\Delta\varphi$,

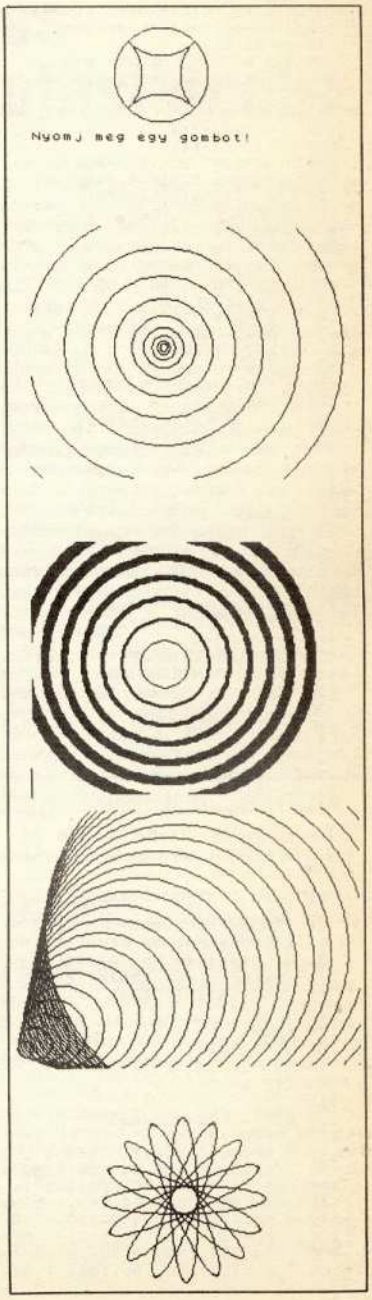
$\cos(\varphi+\Delta\varphi)=\cos\varphi*\cos\Delta\varphi-\sin\varphi*\sin\Delta\varphi$.

2. ábra



3. ábra



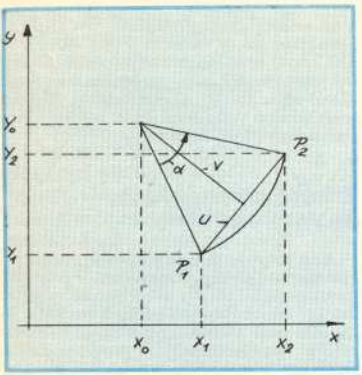


```
 10 {$L-}
 20 {1:KOR filenev alatt hivhato}
 30
 40 PROCEDURE Arc(X1,Y1,X2,Y2:INTEGER;ALFA:REAL);
 50 {X1,Y1,X2,Y2=kezdo es vegpont koordinatai,
 60 R=sugar, ALFA=kozepponti szog radianban}
 70 CONST DI=3;
 80 VAR R,X0,Y0,U,V,DX,DY,
 81     X,Y,XE,YE,DFI,sin,cos:REAL;
 90     I,N:INTEGER;
100 BEGIN
110   DX:=X2-X1;
120   DY:=Y2-Y1;
130   U:=SQRT(SQR(DX)+SQR(DY));
140   R:=U*0.5/SIN(0.5*ALFA);
150   V:=R*SIN(0.5*ALFA);
160   IF ALFA<0 THEN V:=-V;
170   X0:=X1+DX*0.5-DY*V/U;
180   Y0:=Y1+DY*0.5+DX*V/U;
190   DFI:=DI/R;
200   N:=TRUNC(ABS(ALFA/DFI)+0.5);
210   DFI:=ALFA/N;
220   sin:=SIN(DFI);
230   cos:=COS(DFI);
240   XE:=X1-X0;
250   YE:=Y1-Y0;
260   FOR I:=1 TO N DO
270   BEGIN
280     X:=XE*cos-YE*sin;
290     Y:=YE*cos+XE*sin;
300     PixelLine(ROUND(XE+X0),ROUND(YE+Y0),
301     ROUND(X+X0),ROUND(Y+Y0));
310     XE:=X;
320     YE:=Y
330   END
340 END; {Arc}
350
360 PROCEDURE Circle(X0,Y0,R,H:INTEGER);
370 {X0,Y0=kozeppont koordinatai, R=sugar,
371   H=vonalvastagsag}
380 CONST PI=3.141592;
390       DI=3;
400 VAR X,Y,XE,YE,DFI,sin,cos:REAL;
410     I,N:INTEGER;
420 BEGIN
430   DFI:=DI/R;
440   N:=2*TRUNC(ABS(PI/DFI)+0.5);
450   DFI:=PI/N*2;
460   sin:=SIN(DFI);
470   cos:=COS(DFI);
480   XE:=R;
490   YE:=0;
500   FOR I:=1 TO N DO
510   BEGIN
520     X:=XE*cos-YE*sin;
530     Y:=YE*cos+XE*sin;
540     FatLine(ROUND(XE+X0),ROUND(YE+Y0),
541     ROUND(X+X0),ROUND(Y+Y0),H);
550     XE:=X;
560     YE:=Y
570   END
580 END; {Circle}
590
600 PROCEDURE Ellipse(X0,Y0,R1,R2:INTEGER;
601     ALFA:REAL);
610 {X0,Y0=Kozeppont koordinatai,
611 R1=felnagytengely, R2=felkistengely,
620 ALFA=elforditasi szog}
630 CONST PI=3.141592;
640       DI=3;
650 VAR X,Y,XE,YE,XK,YK,XKE,YKE,H,DFI,
651     sin,cos,SINalfa,COSalfa:REAL;
660     I,N:INTEGER;
670 BEGIN
680   H:=R2/R1;
690   DFI:=DI/R1;
700   N:=2*TRUNC(ABS(PI/DFI)+0.5);
710   DFI:=PI/N*2;
720   sin:=SIN(DFI);
730   cos:=COS(DFI);
740   SINalfa:=SIN(ALFA);
750   COSalfa:=COS(ALFA);
760   XKE:=R1;
770   YKE:=0;
780   XE:=H*XKE*COSalfa-YKE*SINalfa;
790   YE:=YKE*COSalfa+H*XKE*SINalfa;
800   FOR I:=1 TO N DO
810   BEGIN
820     XK:=XKE*cos-YKE*sin;
830     YK:=YKE*cos+XKE*sin;
840     X:=H*XK*COSalfa-YK*SINalfa;
850     Y:=YK*COSalfa+H*XK*SINalfa;
860     PixelLine(ROUND(XE+X0),ROUND(YE+Y0),
861     ROUND(X+X0),ROUND(Y+Y0));
870     XKE:=XK;
880     YKE:=YK;
890     XE:=X;
900     YE:=Y
910   END
920 END; {Ellipse}
>L
```

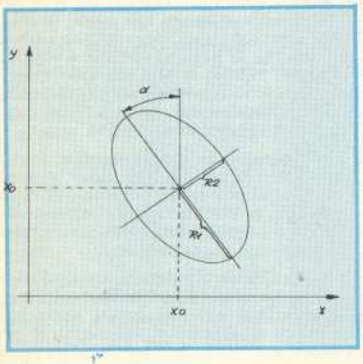
1. lista

2. lista

```
 10 {$L-}
 20 PROGRAM KorDemo;
 30 {Demonstracios program korok, korivek
 31 ellipszisek rajzolasara}
 40 {keszitette Kaboldy Peter 1987. jan.}
 50 VAR I,R:INTEGER;
 60
 70 {$L+}
 80 {$F 1:LINE  }
 90 {$L+}
100 {$F 1:KOR   }
110
120 BEGIN {Foprogram}
130   PAGE;
140   Arc(100,100,100,150,1.57);
150   Arc(100,150,50,150,1.57);
160   Arc(50,150,50,100,1.57);
170   Arc(50,100,100,100,1.57);
180   Arc(100,100,100,150,-1.57);
190   Arc(100,150,50,150,-1.57);
200   Arc(50,150,50,100,-1.57);
210   Arc(50,100,100,100,-1.57);
220   WRITELN('Nyomj meg egy gombot!');
230   REPEAT UNTIL INCH<>CHR(0);
240   PAGE;
250   FOR I:=0 TO 11 DO
260   BEGIN
270     R:=(I*I*I)DIV 10+3;
280     Circle(100,100,R,1)
290   END;
300   PAGE;
310   FOR I:=1 TO 7 DO
320   BEGIN
330     R:=I*15+3;
340     Circle(100,100,R,I)
350   END;
360   PAGE;
370   FOR I:=1 TO 23 DO
380   Circle(I*7,I*4+10,I*6,1);
390   PAGE;
400   FOR I:=0 TO 8 DO
410   Ellipse(122,95,60,10,3.14159/8*I)
420 END.
>C
```



4. ábra



5. ábra

Ezt felhasználva, valamint behelyettesítve a P pont koordinátáit, a következő összefüggéseket írhatjuk fel:

$$X' = (X - X_0) * \cos \Delta\varphi - (Y - Y_0) * \sin \Delta\varphi + X_0$$

$$Y' = (Y - Y_0) * \cos \Delta\varphi + (X - X_0) * \sin \Delta\varphi + Y_0$$

Ha most meggondoljuk, hogy egy kört úgy is származtathatunk, hogy egy pontot egy másik pont körül elforgatunk, akkor ez utóbbi két képlet tulajdonképpen a kör egy egyenletrendszereként is felfogható. Ha felvesszük a kör középpontjának X_0 és Y_0, valamint egy tetszőleges pontjának X, Y koordinátáit, akkor egy $\Delta\varphi$ középponti szögre lévő újabb pontját meghatározhatjuk (X', Y'). Majd ezt tekinthetjük kezdőpontnak és egy újabb pontot határozunk meg és így tovább. Ezzel a számítási idő is csökkenthető, mivel a sin $\Delta\varphi$ és cos $\Delta\varphi$ értékek kiszámítására csak egyszer van szükség.

Az *1. lista* a fenti egyenletrendszer felhasználásával programozott Circle-eljárást használ a kör rajzolására. A DI állandóval azt állítjuk be, hogy a két szomszédos kiszámított pont között mekkora legyen a távolság. Próbálja az olvasó megváltoztatni értékét! A 430—450-es sorok a $\Delta\varphi$ szög pontos beállítását végzik úgy, hogy a rajzolás végén éppen a kiindulópontba jussunk vissza. A kiszámított körpontokat az előző cikkben ismertetett FatLine-eljárással rajzoltattuk ki, de ez természetesen helyettesíthető más egyenesrajzoló algoritmussal.

A körív rajzolása a fentiek alapján egyszerű. Az Arc-eljárás valósítja meg az egyik lehetőséget. Itt a bemenő paraméterek a BASIC-hez hasonlóan a két végpont koordinátái és a középponti szög. Ez esetben ki kell számolnunk előbb a középpont koordinátáit és a kör sugarát, valamint a $\Delta\varphi$ szöget. A viszonyokat a *4. ábra* szemlélteti.

Végül kihasználhatjuk az alkalmat, hogy tulajdonképpen mindent előkészítettünk az ellipszisrajzoló eljárás megírásához. Ellipszist az origó középpontú körből egyszerűen úgy származtathatunk, hogy az eredeti kör minden X koordinátáját egy állandóval megszorozzuk. Így természetesen csak olyan ellipszisek rajzolhatók, melyek nagy és kis tengelye párhuzamos a koordinátatengelyekkel. Ha általános helyzetű ellipszist akarunk rajzolni, akkor a fent ismertetett algoritmus szerint minden pontot egy adott állandó szöggel a középpont körül el kell forgatni.

Ezt az algoritmust valósítja meg az Ellipse nevű eljárás. A bemenő paraméterek értelmezése az *5. ábrán* látható. Végül a *2. lista* egy kis demo programot tartalmaz. Látható, hogy nemcsak a KOR fájlnév alatt kimentett (1. listán bemutatott) forrásnyelvű programrészt kell include-álni, hanem a LINE fájlt is, melyben a pont- és egyenesrajzoló eljárásaink vannak. Csak emlékeztetőül: a 80-as sorban a fájlnév utáni két szóköz nem felesleges, ugyanis lemezmeghajtó esetén kötelező a nyolcbájtos fájlnév használata.

Figyeljük meg, hogy az ismertetett Pascal-eljárásokkal az egész képernyőre rajzolhatunk; ha a kör részben a képernyőn kívülre esik, a program nem áll le. Ugyancsak rajzolhatók körök és körívek vastag vonallal is. A bemutatott eljárások alapján az olvasó is könnyen készíthet olyan eljárásokat, melyek például szaggatott vonallal rajzolnak köröket vagy vastag vonallal ellipszist. A demo program a bemutatott rajzokat rajzolja fel a képernyőre.

DR. KABOLDY PÉTER

Függvények és utasítások I.

A következőkben áttekintjük a magas szintű programozási nyelvek egyik legfontosabb építőkövének, a függvénynek, illetve eljárásnak főbb jellemzőit. Az ember az iskolában hosszú évek során megtanul bizonyos matematikai fogalmakat, például a függvény fogalmát. A programozási nyelvek is kialakítottak maguknak ilyenféle absztrakciós eszközt, a függvény, illetve eljárás fogalmát. Az írás indítéka az a tapasztalat, hogy helytelen dolog összekeverni e két, kétségtelenül rokon fogalmat. Az eljárás nem függvény a szó matematikai értelmében, sőt a programozás függvényfogalma sem azonos a matematika függvényfogalmával, mert fontos szerepe van benne a dolgok időbeni lefolyásának, amivel a matematika nem foglalkozik.

Mindazonáltal, ha olyan függvényeket írunk és használunk, amelyek közel vannak a matematikai értelemben vett függvényekhez, akkor egy sor olyan programozási buktatót elkerülhetünk, amelyekre a programírásnál egyébként állandó figyelemmel kellene lennünk.

(A cikket három részben, havi folytatásokban közöljük. A szerk.)

A problémakör — bevezetés és áttekintés

Minden programozó tudja, hogy a programok apróbb lépésekből tevődnek össze. Nem arra gondolok most, hogy formailag a programot valmilyen egységekre, mondjuk utasássorokra kell bontani, hanem még ennél kisebb egységek is vannak. Ha például leírom azt a BASIC utasítást, hogy A=B*C+A*D, ez is egyszerűbb műveletekre vezethető vissza: B-t szorozd meg C-vel, majd vedd elő A-t és szorozd meg D-vel, az eredményeket add össze, és az így kapott értéket írd az A változóba.

Általában minden nyelvben vannak elemi tevékenységek, és ezek két csoportba sorolhatók: egyesek olyanok, amelyek során valamilyen érték keletkezik, és azonos feltételek között mindig ugyanaz az érték. Ilyen például, ha két változó értékét össze-

szorozzuk; ezt egymás után akárhányszor tesszük is, mindig ugyanazt az értéket kapjuk, amíg a változók értéke nem módosul. Vannak viszont jóvátehetetlen dolgok, amelyek után a program más állapotba kerül, és ha még egyszer elvégezzük azt a bizonyos műveletet, az eredmény nem feltétlenül lesz ugyanaz, mint az előbb. Ilyen például egy adat beolvasása, egy változó értékének növelése stb.

Elméleti szempontból az első csoportba tartozó műveleteket nevezzük függvényeknek, a második csoportba tartozókat pedig utasításoknak. A fenti első csoportba tartozik az összeadás és a szorzás, de ilyen például egy szinusz függvény hívása is BASIC nyelvben. A második csoportba tartozik a változó felülírása, de ilyen például a RND (a véletlenszám-generáló) függvény is, hiszen ez gyakorlatilag minden hívás után más értéket ad.

Ha az első csoportba tartozó tevékenységből többet egymás után végrehajtunk, a helyzet változatlan marad, hiszen a program állapota nem változik; azaz ezt a tevékenységsort egy nagyobb összetett függvénynek tekinthetjük. Hasonlóan: több utasítás és függvény egymás utáni végrehajtását egy nagyobb összetett utasításnak tekinthetjük.

A programozási nyelvek különbözősége

Az, hogy egyszerű tevékenységek egy (gyakran használt) sorozatát valamiképpen összefogjuk és azt a későbbiekben egyetlen tevékenységnek tekintve használjuk, a programírás egyik legalapvetőbb fogása. Erre minden programozási nyelv lehetőséget ad. Arra azonban, hogy az így keletkezett új eszközt el is nevezzük, és ezentúl egyszerűen a név leírásával hivatkozzunk rá, már nem minden nyelv ad módot. Nem ad ilyen lehetőséget egy sor gépi kód sem, egy sor assembly nyelv és a BASIC nyelv számos variánsa sem (Sinclair ZX81, HT—1080Z, az Atari BASIC nyelve stb.). Sok más BASIC változatban ez a lehetőség meglehetősen korlátozott: általában csak az egyetlen utasításból álló DEF FN-re van lehetőség (Sinclair Spectrum, a különböző Commodore gépek, az IBM PC Microsoft BASIC stb.). Ennek az absztrakciós képességnek a hiánya miatt hívják ezeket a nyelveket alacsony szintű nyelveknek.

Egy összetett tevékenységet — ha az állapotot nem változtatja meg — ezentúl (összetett) függvénynek, az állapotot megváltoztatót pedig eljárásnak (angolul: procedure) fogjuk nevezni. Ez a megkülönböztetés az elméleti kérdésekkel foglalkozók számára elfogadott. Sajnos nem ez a helyzet a programozási nyelveknél, ahol néha mindent függvénynek hívnak (például C nyelv), néha mindent eljárásnak (ALGOL 60, ALGOL 68, Modula—2). Néhány nyelv megkülönbözteti a függvényt és az eljárást (például a FORTRAN és a Pascal) hasonló, de nem pontosan ilyen értelemben.

A magas szintű nyelvekben kialakult a konvenció arra nézve, hogyan kell egy ilyen függvényt, illetve eljárást definiálni, és hogyan kell a végrehajtását meghívni.

Függvénydefiniálás és -hívás

A függvénynek először megadjuk a nevét, majd zárójelek között felsoroljuk a szükséges paramétereket, megadva azt, hogy azokhoz milyen típusú (egész, valós, karakter, sztring stb.) érték tartozik. Végül megadjuk, hogy a függvény milyen típusú értéket ad vissza. Ezután következik a függvény törzse: az a tevékenységsorozat, amelyet a függvény nevében végre kell hajtani. A paramétereket el kell neveznünk — bizonyos formális nevekkel —, hogy a törzs utasításai hivatkozhassanak rájuk, például:

Körterület(sugár:REAL):REAL=sugár*sugár*3,1415

A függvény hívása ezek után: leírjuk a függvény nevét, majd zárójelben utánateszszük értékének egy olyan sorozatát, amelyet a paraméterekhez akarunk rendelni. Az értékeket kifejezésekkel adhatjuk meg, legegyszerűbb esetben változókkal és konstansokkal, bonyolultabb esetben függvényhívásokkal, amelyek egyaránt lehetnek beépített függvények (mint az összeadás vagy a szorzás), de természetesen lehetnek a felhasználó által megadott függvények is. A függvényhívás eredményül egy értéket ad vissza, és olyan helyeken használható fel, ahol csak egy értékre van szükség. Példa erre egy olyan henger felszínének kiszámítása, amelynek magassága azonos az átmérőjével, a fenti **Körterület** és a **Palást**(sugár,magasság:REAL):REAL függvény segítségével:

Hengerfelszín=2***Körterület**(R)+**Palást**(R,2*R)

Eljárásmegadás és -hívás

Az eljárások megadásának nagyon hasonló a formája, de tartalmilag lényeges eltérések vannak. Mivel az eljárást arra „találtuk ki" — mondhatni: az eljárás azért van —, hogy megváltoztassa a program állapotát, ezért az eljárásnak lehetősége van a változók kiolvasására és megváltoztatására, tehát tulajdonképpen semmi szükség nincs rá, hogy a nevén kívül bármi mást is megadjunk paraméterként. Azzal azonban, hogy mégis paramétereket vezetünk be, két dolgot érünk el: egyrészt rugalmasabbá tesszük az eljárás felhasználhatóságát, másrészt láthatóbbá tesszük, hogy mit és hogyan módosítsunk. Ezért az eljárásoknál is szokásos egy paraméterlista bevezetése az eljárás neve után, amelyben bizonyos dolgokat felsorolunk. (Ebben a felfogásban az eljárás nem ad vissza értékeket, hiszen utasítás jellegű valami.)

A paraméterlistában kétféle dolog lehet: változók és értékek. A változó paraméter megadja, hogy ennyit vagy annyit, így vagy úgy módosít. Természetesen a két paramétertípust meg kell különböztetnünk egymástól. Például: egy változóban egy kétjegyű számot tárolunk, amelyet valamilyen értékkel megnövelünk: ha túllépjük a 99-et, a felső számjegyeket elhagyjuk.

```
INC(VAR:a:INTEGER; VAL:k:
                    INTEGER)
BEGIN a:=a+k; a:=a MOD 100 END.
```

Az elnevezések: INC=increment (azaz növeld) VAR=variable (azaz változó), VAL=value (azaz érték). Az INC(A,3) eljáráshívás az A változót 3-mal növeli, az INC(A,A) hívás az A változót saját értékével megnöveli, azaz megduplázza. Ezzel szemben az INC(3,A) eljáráshívás illegális, mert az első változóval jelölt argumentumban nem változó neve áll.

Előnyös tulajdonságok és viszonylagosságuk

Az a tény, hogy a függvények csak a paraméterlistájukban megadott értékeket használhatják föl, és eredményül egyetlen értéket adnak, közben pedig a program állapotát nem változtatják meg (a változók értéke nem módosul, a program futása ugyanott folytatódik), meglehetős áttekinthetőséget kölcsönöz azoknak a programoknak, amelyek ilyenekre építenek. Ha ránézünk egy ilyen programra, amely függvényhívásokat tartalmaz, még ha nem is tudjuk egész pontosan, hogy mit csinál a függvény, azért azt látjuk, hogy milyen adatokból indul ki, és hogy mi történik a végeredménnyel. Például a fent szereplő henger felszínének kiszámítása valószínűleg mindenkinek érthető volt, noha a **Palást** nevű függvényt egyáltalán nem adtuk meg.

A programozás módszertanával foglalkozó könyvek azt ajánlják, hogy törekedjünk az eljárások esetén is hasonló áttekinthetőségre, és ezért az eljárás paraméterlistájában soroljunk fel mindent, amit az eljárás felhasznál vagy módosít.

Ezt az ajánlást a programozási nyelvekbe be is lehetne építeni, de hatékonysági megfontolásokból egyetlen ismert nyelv sem tesz ilyen megkötéseket. Képzeljük el például, hogy van egy levelünk, amelyet számtalan embernek el akarunk küldeni. Könnyű írni egy olyan programot, amely beolvas egy listát a címzettekről, és a szövegbe adott helyekre beírja az illető nevét. Programunk valószínűleg három részből fog állni: először beolvasunk egy nevet a listáról, majd beleírkáljuk ezt az adott helyekre, végül a kész levelet kinyomtatjuk. Mindezt addig ismételjük, amíg a lista tart. A középső rész egy olyan eljárás lehet, amely egy nevet kap paraméterként, és beírja az adott helyekre.

Ez az eljárás természetesen nem felel meg a fenti elvárásoknak, hiszen implicite felhasználja a szöveget és a pozíciókat, ahova a nevet be kell írni. A korrekt megoldás egy olyan általános eljárás lenne, amely paraméterben kapná meg a szöveget, a kitöltendő helyek listáját és a beírandó szöveget. Csakhogy az általánosságnak ára van, az általános eljárás valószínűleg jóval bonyolultabb lenne, mint a speciálisan egy levélre alkalmazható, és az eljárás meghívása is bonyolultabb lenne. Ilyenkor sokakban felmerül a gondolat, hogy nem egyszerűbb dolog-e a speciális programot igazítani hozzá az újabb és újabb levelekhez, mint az általánosat megírni és esetről esetre hasonló nagyságú munkával felparaméterezni.

FARKAS ERNŐ

Cambridge Computer Z88 Az új Sinclair-csoda

A számítógép-rajongók körében legendás Sir Clive Sinclair, az angol lovag ismét hallat magáról. A hazai ZX81-, Spectrum- és QL-tulajdonosok tizenötezres tábora bizonyára szívesen cserélné fel gépét a Sinclair-cég legújabb darabjával.

Íme lássuk a medvét, pontosabban a majmot, a brit Personal Computer World márciusi számának címlapjáról: csomagtartójában a Z88-cal motorozik a jövőbe. Állati egyszerű, mondaná, ha tudna beszélni. Közlendője valahogy így hangzana: — Végy egy Zilog Z80 mikroprocesszor-morzsát, egy közönséges ULA-t, egy 32 k-s DRAM-ot meg egy 1 Mbites EPROM-ot. A négy chipet néhány egyéb tranzisztorral, ellenállással, kondenzátorral és más egyszerű eszközzel kiegészítve forraszd a nyomtatott áramköri panelra. Mindezt építsd be az A4-es lapméretű — a billentyűzetet és a kijelzőt tartalmazó — dobozba. Végül töltsd fel a hétköznapi munkákhoz használható alkalmazási programokkal, és már kész is a számítógép. S ha csak egy kicsit is olcsóbb, mint versenytársai, a népszerű konstruktőr csodamasinája komoly üzleti sikerre számíthat az idei karácsonyi vásáron.

Sir Clive Sinclair tizenöt esztendeje dobta piacra a világ első programozható zsebszámológépét. A 24 programlépést tároló, mindössze 40 angol fontért árult géppel már a később kiteljesedő Sinclair-üzletpolitikát bizonyította. Olyan termékkel tört be a piacra, amelyhez hasonlót ugyan a „nagyok", a piac diktátorai jó minőségben, kiterjedt alkalmazásokkal és szolgáltatásokkal forgalmaztak már, de magas áron. Sinclair olcsóbban, gyengébb minőségben, kevésbé bő szolgáltatásokkal, ám a tömegek számára megvásárolható termékkel jelentkezett. Ennek az üzletpolitikának igazán nagy sikert a ZX81 és a Spectrum-sorozat hozott. Ekkor már meg kellett küzdeni a hasonló szellemben dolgozó versenytársak termékeivel, mint a Commodore VIC-20-szal vagy a Tandy CoCo-val. Ezek tudták mindazt, amit Sinclair gépei, és áruk sem volt magasabb. A nagy üzlethálózatok is szívesen árulták ezeket a keresett gépeket.

A QL (Quantum Leap, azaz kvantumugrás) viszont csődbe „ugratta" a Sinclair céget. Az

A Z88 a „Personal Computer World" címlapján

akkorit, ugyanis a lovagnak több vállalkozása is volt. Miért ment mégis tönkre? Mert eltért az addig jól bevált üzletpolitikától. A QL az árához képest keveset nyújtott, nem volt olcsó, alkalmazási és segédprogramjai pedig egyenesen siralmasnak mondhatók.

A Z88 integrált szoftverű, igen sokoldalú személyi számítógép. Súlya alig egy kilogramm, vastagsága mindössze két centiméter, vagyis bármely szabványos kézitáskában kényelmesen elfér. Négy darab, sétálómagnóban használatos elemről működik, melyek húszórás üzemidőre elegendők, vagy ha nem használjuk a gépet, akár egy évig is eltartanak. (Ne lepődjünk meg, hiszen a RAM-ot állandóan frissíteni kell, és ez némi fogyasztással jár.)

Az Epson gyártmányú, kristályfolyadékos (krifó) kijelző nyolc sorban, soronként száz karaktert mutathat. A nyolcvan karakter széles fő ablak mellett bal oldalon a menü (106 karakter), jobb oldalon egy egész oldalnak a térképe látható. Ez utóbbi mező a szöveg 64 sorát mutatja. Minden egyes képpontnak egy karakter felel meg. A jobb felső sarokban a menüből kiválasztott funkciót láthatjuk.

A konstrukció külön érdekessége az írógépszerű billentyűzet, amelyet teljes egészében egy gumipelenka takar. Nyilván azért, hogy ne kerülhessen szennyeződés a finom szerkezetbe. Így a klaviatúra — és a gép is — teljesen hangtalanul működik, tehát tanórán, értekezleten vagy akár társaságban is dolgozhatunk vele, anélkül, hogy másokat zavarnánk.

A gép elülső szélén a tárbővítő csokoládék számára három perselynyílás van. A kétszer két centiméteres, két centi vastag — a Microdrive kazettákhoz hasonló — külső tárakat egyelőre 32 és 128 k-s méretben kínálják. Az elméletileg még kezelhető, legnagyobb tárkapacitás 4 Mbájt, de belátható időn belül csak az egy Mbájtost ígérik. A külvilággal a kapcsolatot egy RS232 csatlakozó nyújtja: a kilencpólusú soros port átviteli sebessége legfeljebb 1200 baud lehet. A beépített szoftver az FX típusú soros nyomtatók kezelésére alkalmas. Külön kapható kábel és floppy szolgál az IBM vagy azzal kompatibilis PC-vel folytatható adatcserére.

A beégetett alkalmazási programok az átlagember mindennapi igényeihez igazodnak: négy alapműveletes kalkulátor, naptár, pontos idő, határidőjelző, kartoték, napló, BASIC és ... és VALAMI EGÉSZEN CSODÁLATOS, a tervezők által szerényen csak „Vágyálom"-nak (Pipedream) keresztelt alkalmazási program. Nos, a Vágyálom tényleg mindent tud, amire valaki álmában csak vágyhat. Szövegszerkesztő, táblázatkezelő és adatbázis — egyszerre, egyben. Milyen kár, hogy csupán olvastunk és hallottunk róla, de a gyakorlati tapasztalatokról már nem számolhatunk be.

Az ára? Forgalmi adó — VAT, Mehrwertsteuer — nélkül 250 font (MNB-árfolyamon 19-20 ezer forint), tehát nekünk sem elérhetetlenül sok. A 32 k-s tárbővítő 13, a 128 k-s 43 fontba kerül.

Mikrogépek,

Kinek milyen. Van, akinek gazdag ajándékokat hoz, utána kellemes emlékek maradnak, míg másoknak szegényesebb, szolídabb. Vannak karácsonykor boldogok, és vannak szorongók, magányosak. Ahány család, annyiféle karácsonyi program. Ami közös, az a faállítás és az ajándékozás. Addig azonban végig kell járni a vásárlás stációit, elviselni a nyűgét, eldönteni, hogy pénztárcától függően ki milyen meglepetést szerezzen hozzátartozóinak. És akkor még egyáltalán nem biztos, hogy az elhatározás nyomán megtalálja az elképzelt ajándékot az üzletben, vagy hogy tudja az ajándékozó, mire hol lel.

Itt most azoknak kívánunk segíteni, akik ifjabb vagy idősebb szeretteiknek számítástechnikai berendezést, ahhoz kiegészítőt, illetve programokat akarnak ajándékozni. Szeptemberi körsétánkon felkerestünk néhány üzletet, ahol ugyan pontos adatokkal nem szolgálhattak e termékek áráról, sem mennyiségéről, de körülbelüli összegeket azért mondtak. Természetesen ajánlásunkban nem törekedhettünk a teljességre. Arra viszont igen, hogy megkíséreljünk metszetet adni arról, hogy mi és milyen áron kapható a boltokban.

Az V. kerület Múzeum körút 19-ben található a Foto Elektronik reprezentatív üzlete. *Hrabovszky Tibor* boltvezető készségesen adta a következő felvilágosítást.

— A Commodore számítógépcsaládból Plus/4-et, C16-ot és C64-et megfelelő mennyiségben kínálunk, és van választékunk ezekhez kapcsolható perifériákból is. A Plus/4 magnóval együtt 18 ezer, a C16 magnóval együtt 10—11 ezer, a C64 ugyancsak magnóval 24—26 ezer forint lesz. Ajánlunk ezekhez a berendezésekhez lemezmeghajtót (az 1541—es típus ára 25—28 ezer forint), továbbá nyomtatókat. Ezek közül az MPS—803-ast 25 ezer forintért, a Citizen 120D típusút pedig minőségtől függően 38—42 ezer forintért. Monitorokat is árusítunk a karácsonyi nagy vásárláskor: a színesek ára 40 ezer forint lesz, a Philips márkájú monokrom 16—18 ezerbe kerül.

A vastagabb pénztárcájúaknak szól a következő ajánlat. Eredeti IBM PC XT és AT — kiépítéstől függően — 500 ezer, illetve 850 ezer forinttól kapható. A kompatibilis XT ára 150 ezer forintnál, az AT-é 300 ezernél kezdődik. Mind a két típushoz ugyanazok a perifériák használhatók. Ennek megfelelően kínáljuk az EPSON

FX—1000—es nyomtatót 115—130 ezer forint körüli áron, a PLOTTER HP7475A-t 400 ezer forintért. Ez utóbbi A/3-as formátumban, hat színben rajzol.

Üzletünkben felkészültünk a különböző winchesterek és streamerek (például 10-től 80 Mbájtosig) és különféle számítógép-tartozékok, többek között nyomtatókhoz szükséges festékszalagok (800—1500 forint), floppyk (10 darabos dobozonként 1000—4000 forint), kazetták árusítására. Forgalomba hozunk számítógépes részegységeket is: például az IBM PC XT-hez kártyákat, vázat, tápegységet.

A Centrum-hálózat karácsonyra kínált termékeiről *Pásztor Tamás* rendszerszervező számolt be részletesen.

— Változatlanul árusítjuk — és remélhetően lesz belőle elegendő — az Enterprise mikroszámítógépet, 16 900 forintért, és ismét forgalomba hozzuk hozzá a korábban nagy sikert aratott és gyorsan szétkapkodott 15 legkeresettebb játékprogramot. Slágerlista alapján állítottuk össze a leginkább hiányolt játékok listáját, és ezeket igyekszünk pótolni. Újra kaphatók lesznek például a Sorcery, az Autóverseny, az Orient-express, a Beach-head és a sakk (Syrius II) kazetták. Újdonságként 10 további játékprogrammal jelentkezünk a piacon, amelyeket reméljük, tetszéssel fogadnak majd az Enterprise-tulajdonosok. Ezek közül csak példaként sorolom fel a Digitális pókot, a Newton almáját és a Hungaroring Forma—1 autóversenyt.

A fordítóprogramok árusításával bővíteni kívánjuk az Enterprise-hoz kapható programok körét. Így vásárolható majd Assembler, LISP, Pascal, illetve E-FORTH program. És ami bizonyára tetszést arat: üzleteinkbe kerül az a speciális program, amellyel a Commodore perifériákat csatlakoztatni lehet az Enterprise-hoz. Ilyen lesz a VC—1541-es floppyhoz és az MPS—801-es mátrixnyomtatóhoz alkalmazható program. E termékünk árát ma még nem tudjuk. Azt viszont már igen, hogy háromkötetes sorozattal jelentkezünk. A Hetedhét Enterprise című könyvek — EXDOS 2.1, EXDOS 1.3 és ISDOS 1.0 — 150 illetve 300 forintért kaphatók majd. Katalógust, programozási segédletet, ötleteket, a géphez trükköket és forráslistákat tartalmaznak. Végül különböző kiegészítőket és tartozékokat ajánlok az Enterprise számítógéphez: monitor-csatlakozókábeleket 648 forintért, RS232C interfészkábelt 648 forintért, s ugyanennyiért Centronics interfészkábelt nyomtatóhoz. Az Enterprise—Sinclair mikrokapcsoló ára 618 forint. Segítségével ki- és bekapcsolhatóvá tehető a számítógép.

A hobbi kategóriájú DATACOOP PRT—42 BABY típusú mátrixnyomtató kábellel együtt 22 950 forintba, a professzionális EPSON—RX 80-as mátrixnyomtató 49 950 forintba kerül. Mindkettő a szabványos magyar ábécé betűit is nyomtatja.

A VII. kerület, Tanács körút 3/A-ban a nagy műszaki bizományi áruházban *Martinov László* üzletvezető és helyettese, *Erki Imre*

„Kis karácsony, nagy karácsony...

makro-árak

adott tájékoztatást. Mindketten elmondták, hogy a szeptemberi adatok nem véglegesek, csupán informatív jellegűek.

— A Sinclair ZX 128 k-s berendezés a cég legújabb terméke. Mi 33 ezer forintért árusítjuk. A C64 régebbi típus, így ára várhatóan csökken, és 25—30 ezer forintért kapható majd. A Commodore 128 ma még drágább, de karácsonykor valószínűleg 55 ezer forint lesz az ára. A perifériákból a következőket ajánlhatjuk. DATASETTE magnó — a C64-hez — eredeti és hazai gyártmány 4 ezer forint körüli áron kapható. A Sinclair ZX 128 k géphez kínálunk kis lemezmeghajtót 10 ezer forintért, és ugyanennyiért árusítjuk az Interface I-et. Új típus a 128 k-s Plus 2-es Sinclair gyártmányú, beépített normál kazettás magnóval rendelkező számítógép 35 ezer forintért.

Commodore gépekhez 1541-es lemezmeghajtó 30 ezerért, illetve a 1541C típus 35 ezer forintért vásárolható nálunk. Ugyanehhez a számítógéptípushoz nyomtatóink is vannak: az MPS—803-as 33 ezer forint, a Citizen 120D 38 ezer, a Seikosha SP 180 szintén 38 ezer forintba kerül. A Panasonic KXP 1090i típusú nyomtató 45 ezerért, az MPS—802-es típus 49 ezerért kapható. Természetesen valamennyi nyomtatónkhoz festékkazettákat is forgalmazunk, 1—2 ezer forint közötti áron.

A floppyválaszték bőséges. Általában 10 darabos csomagolásban kerülnek forgalomba, de darabonként is megvásárolhatók. Egy-, illetve kétoldalas dupla sűrűségű floppyk 700—1500 forintól kaphatók. Az ugyanilyen tulajdonságú márkásak viszont 2—4 ezer forintba kerülnek. A márkás 3M-es típus 3,5", egyoldalas, dupla sűrűségű, darabja 800 forint. Széles skáláját kínáljuk a különböző kazettáknak, játékprogramoknak, 300—1000 forint közötti áron. A Quick Shot I., II., IX., és X. botkormánytípusok ára 1000—3000 forint.

Az IBM PC XT és AT gépek eredeti kiépítéstől függően 500 ezer forinttól kaphatók. Az IBM PS/2 modell 30 (a legújabb berendezés) 900 ezer forint. Az IBM PC XT/AT kompatibilis gép — 640 k, multiprogramozhatóság, monokrom monitorral, floppymeghajtóval (360 k) — 105 ezer forint, s ugyenehhez a 20 Mbájtos winchester 60 ezer forint. Az IBM PC-hez kapható kompatibilis Philips monitorok ára 50 ezer forint.

A nyomtatók közül a Mannesmann-Tally MT80 PC-t ajánljuk 49 ezerért és az EPSON FX—1000-et 114 ezer forintért. A Canon PC5 típusú fénymásoló ára 150 ezer forint.

A Skála Metró Kereskedelmi Leányvállalat Metro-Comp boltjában, a VI. kerület, Jókai utca 40-ben *Hatlaczki Tibor* osztályigazgató a következőket mondta karácsonyi kínálatukról.

— Számítástechnikai termékeink közül az IBM PC XT/AT kompatibilis gépek, kiegészítőktől függően 140 ezer forinttól kaphatók. Az ezekhez való nyomtatók választékából ajánlom az EPSON FX—1000-et (130 karakteres) 125—130 ezer forintért, illetve a Citizen 120D-t (kisebb írásszélességű, vagyis 80 karakteres) 40 ezerért.

A C64-et várhatóan 15 ezer (!) forintért hozzuk forgalomba a karácsonyi vásárlás idejére. Ehhez a perifériák sem lesznek drágák, hiszen a floppymeghajtó 16—20 ezerbe, a Philips színes monitor 35 ezerbe, a Commodore nyomtató pedig — az MPS—803-as típus — 28 ezerbe kerül majd. A Commodore 128 ára 49 ezer, a hozzá tartozó 1571-es floppymeghajtó szintén 49 ezer, a Citizen 120D (80 karakteres) 40 ezer forint lesz. A termékekre általában fél, illetve egy év garanciát vállalunk.

Alkatrészeket is árusítunk. Kaphatók majd az IBM PC XT-hez és az IBM PC AT-hez gépépítéshez alkalmas alkatrészek, melyek ára mintegy 10-15 százalékkal olcsóbb, mint a kész terméké.

Testvérboltunkban, a hozzánk közel eső Metro-Printben (V., Bajcsy-Zsilinszky út 78.) hasonló termékeket hasonló árakon hoznak forgalomba.

A Novotrade Octa Soft és a Novotrade Delta Soft új játék- és oktatóprogramokkal kedveskednek a számítógép-tulajdonosoknak. Ezekről elöljáróban annyit, hogy áruk a cikk írásakor még nem volt végleges.

A Newton almája rejtvény játékprogrammal — 360 forint — valamennyi Commodore-on, Spectrumon és Videoton TV Computeren és Enterprise-on szórakozhatnak tulajdonosaik. Külön érdekessége ennek az újdonságnak, hogy programmal nyerni is lehet!

A Viktor a piktor — 250 forint — játékkal 101 labirintus útvesztő járható be, s ezek falát festegetni, színezni is kell. Szegény Viktort kóborlása során szellemek üldözik . . . A program Plus 4-es és C16-os mikroszámítógépen futtatható. Az Elvarázsolt kastély — 350 forint — Plus 4-re íródott, s azoknak ajánljuk, akik szeretik az izgalmas, kalandos játékokat. A Water Polo — 450 forint — C64-es géphez használható. A vízilabda kedvelőinek nyújt elsősorban élvezetes szórakozást. A Kocka póker — 340 forint — Plus 4-re íródott, s a hagyományos számítógépes változataként játszható.

Néhány oktatóprogramot is ajánlunk. Így például a nyuszi olvasni tanít — 950 forint — programot, melyet Plus 4-re és C64-re készítettek. A 6 kazettából álló csomaggal az alsó tagozatosokhoz kívánják közelebb vinni az olvasást. Szintén Plus 4-es illetve TVC számítógép tulajdonosok vehetik használt az Optika — 290 forint — oktatóprogramnak. Ez a szoftver a felső tagozatos fizikaórák szemléltető eszköze lehet. A 3 kazettából álló Gyorsolvasó programot Plus 4-re — a programcsomag 500 forint — és C16-ra — 420 forint a 3 kazetta — írták. Ígérik kiötlői, hogy 3 hét alatt a programmal az olvasás sebessége és megértési szintje javítható.

A Videoton TV Computer tulajdonosoknak készült a következő 2 program. Az öt labdajáték — 350 forint — a sport rajongóinak lesz üdítő szórakozás. A Bűvös négyzetek — 280 forint — oktatóprogram az általános iskola 3. osztályától segít a matematika útvesztőiben eligazodni.

És végül egy különleges csemegét ajánlunk. Az irodai munka könnyítésére készítették a Deltex Plus-t — 2500 forint —, mely a Robotron írógépekhez illeszthető szövegszerkesztő cartridge.

Reméljük, mindenki megtalálja az ízlésének és pénztárcájának legmegfelelőbbet.

KRASZNAI ÉVA

Szoftver

Cikksorozatunkban először a gépi kódú programozást segítő három legfontosabb eszközzel, az EDITOR, az ASSEMBLER és a MONITOR programmal kapcsolatos alapvető ismereteket mutatjuk be. Aki gépi nyelven akar programozni, annak ezekre a programokra feltétlenül szüksége van.

Az ASSEMBLER

Az ASSEMBLER-ekről általánosságban

Az ASSEMBLER név magyarul összeállítót jelent, de a magyar számítástechnikai szaknyelvben az eredeti angol elnevezés honosodott meg.

Az ASSEMBLER feladata, hogy a felhasználó által megírt assembly szintű forrásprogramot gépi kódokat tartalmazó tárgyprogrammá alakítsa át. Ezt az átalakítást (fordítást) igen gyakran alkalmazott szakkifejezéssel assemblálásnak is hívják.

Az assembly szintet az ún. gépi szinttől az különbözteti meg, hogy a gépi szintű programozás során az utasításokat és operandusaikat hexadecimális (bináris) formában kell megadni, az assembly szinten írott forrásprogramban pedig az utasításkódok és operandusok helyén szimbólumokat — mnemonikokat — alkalmazhatunk, amelyek hexadecimális (bináris) megfelelőit az ASSEMBLER belső táblázatai tartalmazzák.

Az ASSEMBLER tehát az assembly szintű forrásprogram fordítását hajtja végre, melynek eredményeként létrejön a tárgyprogram. Mint említettük, az assembly szintű forrásprogram szimbólumokat is tartalmazhat. Az ASSEMBLER-ek ezenkívül általában megengedik, hogy az operandusok megadásakor különböző számrendszerekben (bináris, oktális, decimális, hexadecimális) megadott numerikus konstansokat, karakteres konstansokat (sztring = karakterfüzér), egyszerű aritmetikai operátorokat (+, −, *, /), illetve az előbbi elemekből képzett kifejezéseket adhassuk meg.

Az ASSEMBLER működése ún. menetekre tagolódik. Az ASSEMBLER minden egyes menet végrehajtása során végigolvassa a forrásprogramot. Aszerint, hogy a fordítást hány menetben hajtja végre, egy menetben fordító (egymenetes) és két menetben fordító (kétmenetes) ASSEMBLER-ről beszélhetünk.

Itt nem részletezett, számos ok miatt általában kétmenetes ASSEMBLER-eket alkalmazunk. Az alábbiakban vázlatosan ismertetjük az egyes menetek feladatait.

1. menet

A menet végrehajtása során az ASSEMBLER — miközben végigolvassa a forrásprogramot — felépíti az ún. felhasználói szimbólumtáblázatot. E táblázatba kerülnek mindazok a szimbólumok, amelyek nem szerepelnek az ASSEMBLER ún. állandó szimbólumtáblázatában.

Az ASSEMBLER két táblázattal dolgozik: az állandó szimbólumok és a felhasználói szimbólumok táblázatával. Az állandó szimbólumok táblázata egy — az ASSEMBLER indításának pillanatától meglévő — a működés során változatlan táblázat. A gépi utasítások, direktívák (lásd később), regiszterek nevének mnemonikjait (szimbólumait) és a hozzájuk rendelt értéket (gépi kódot) tartalmazza.

A felhasználói szimbólumok táblázata a forrásprogram végigolvasásakor az ASSEMBLER által felismert mindazon szimbólumokat tartalmazza, amelyek nem találhatók meg az állandó szimbólumok táblázatában. A felhasználói szimbólumtáblázat kezdetben üres, de miközben az ASSEMBLER a forrásprogramot olvassa, egymás után felismeri azokat a szimbólumokat, amelyeket a programozó valamely cím vagy adat szimbolikus azonosítására választott — ezek a felhasználói szimbólumok —, és ezeket felveszi a felhasználói szimbólumok táblázatába. Ha ezzel egy időben az ASSEMBLER az adott szimbólum értékét is meg tudja állapítani — és a táblázatban rögzíti is azt —, a kérdéses szimbólumot pre-definitnek (előre definiáltnak) nevezzük. Ha a táblázatba rögzítés során a szimbólum értéke az ASSEMBLER előtt még ismeretlen — és majd csak később fogja tudni megállapítani és rögzíteni —, akkor post-definit (utódefiniált) szimbólumról van szó.

A felhasználói szimbólumok értékeinek megállapítására az ASSEMBLER ún. helyszámlálója (location counter) szolgál. A helyszámláló a fordítás megkezdésekor kezdőértéket kap, majd a forrásprogram fordítása során értéke az éppen fordított utasítás hosszával megnő. Így például ha a helyszámláló pillanatnyi értéke 1564 és az éppen fordított utasítás 3 bájt hosszú, a helyszámláló értéke 1567 lesz.

A forrásprogramban gyakran alkalmazunk ún. címkéket. A címke a forrásprogram valamely pontjának (címének) szimbolikus neve. A fentiek szerint a címkékhez rendelt érték tehát nem más, mint a helyszámláló aktuális értéke a kérdéses szimbólum definiálásának pillanatában.

Miután az ASSEMBLER a forrásprogramot végigolvasta és felépítette a felhasználói szimbólumok táblázatát, befejeződik az 1. menet. Ennek során tehát lényegében csak a szimbólumtáblázat épül föl, kódgenerálás nincs.

A felhasználói szimbólumtáblázat létrehozására azért van szükség, hogy az ASSEMBLER a gépi kód összeállításakor a szimbolikus operandus helyére — a táblázatból kiolvasott — megfelelő értéket helyettesíthesse. Ha azonban ez a szimbólum post-definit, akkor a szükséges szimbólumérték hiányában a kódgenerálás nem hajtható végre. Ezért biztosítani kell, hogy minden szimbólum pre-definit legyen addigra, amikor a kódgenerálás megkezdődik. Ezt azáltal lehet megvalósítani, hogy az 1. menet során az ASSEMBLER létrehozza a felhasználói szimbólumtáblázatot.

Azokat a szimbólumokat, amelyek az 1. menet végére sem válnak pre-definitté (értsd ezen, hogy nem kapnak értéket), nem definiált szimbólumnak (undefined symbol) nevezzük.

2. menet

A 2. menetben készül el a tárgyprogram. Ennek során az ASSEMBLER ismét végigolvassa a forrásprogramot, és azt sorról sorra átalakítja a kérdéses sorban álló assembly szintű utasításnak megfelelő gépi kódra.

Az adott sor értelmezésekor az ASSEMBLER képes különféle szintaktikai (formai), illetve szemantikai (értelembeli, logikai) hibák felismerésére. A hiba észlelésének tényét az ASSEMBLER hibajelzés formájában közli a felhasználóval, majd megkísérli a forrásprogram további fordítását annak érdekében, hogy a program esetleges további hibáit is feltárja. Ezt abban az esetben képes megtenni, ha az előfordult hiba nem ún. fatális hiba. Fatális hiba fellépésével ugyanis a további fordítás már nem lehetséges. (Ilyen fatális hiba például az, hogy a program olyan sok felhasználói szimbólumot tartalmaz, amelyet az ASSEMBLER táblázat korlátozott mérete — vagyis a RAM korlátozott mérete — miatt már nem képes tárolni.)

Az a tárgyprogram, amely egy hibás forrásprogramból keletkezik, értelemszerűen nem tekinthető hibátlannak, és ezért általában nem használható.

Az ASSEMBLER-ek kapcsán gyakran beszélnek az ún. 3. menetről. Ez a menet lényegében nem más, mint a 2. menet, de cél-

ja nem a tárgyprogram, hanem az ún. fordítási lista létrehozása. Ez tulajdonképpen a forrásprogram másolata, amely azonban sorról sorra tartalmazza a helyszámláló aktuális értékét és az adott sorból nyert gépi kódot, valamint hibás sor esetén a hibára jellemző hibajelzést. A fordítási lista tehát lényegében az elvégzett átalakítás dokumentuma. A fordítási listák mindezek mellett általában tartalmazzák a felhasználói szimbólumtáblázat tartalmát névsorba rendező, az egyes szimbólumokhoz rendelt értékekkel kiegészített listát.

Ha a keletkező tárgyprogram és a fordítási lista tárolására két különböző perifériát alkalmazunk — például lyukszalaglyukasztót és sornyomtatót —, a 2. és 3. menet feladatait az ASSEMBLER egyszerre, egy menetben képes végrehajtani.

Az ASSEMBLER-ek a gépi utasítások mnemonikjainak értelmezésén túl még néhány, az ASSEMBLER működését közvetlenül vezérlő ún. direktíva (más néven pszeudo-utasítás) felismerésére és végrehajtására is képesek. Ezek a direktívák általában olyan feladatok elvégzésére utasítják az ASSEMBLER-t, amelyeknek hatása a tárgyprogramba már nem kerül át, tehát csak az ASSEMBLER működését befolyásolja. Van azonban néhány direktíva, amely a tárgyprogramra is hatást gyakorol. A direktívákat funkciójuk szerint általában a következő csoportokba sorolhatjuk:

— a helyszámláló értékét beállító direktívák (ORG, DEFS),

— szimbólumdefiniáló direktívák (EQU, DEFL),

— a tárgyprogramba konstans értéket elhelyező direktívák (DEFM, DEFB, DEFW),

— a fordítás befejezését elrendelő direktíva (END),

— feltételes fordítást vezérlő direktívák (IF, ENDIF),

— programszegmentálást, relokálást szervező direktívák (GLOBAL, NAME, PSECT),

— MACRO-ASSEMBLER esetén a makrók szervezését vezérlő direktívák (MACRO, ENDM),

— fordítási lista formátumát vezérlő direktívák (EJECT, TITLE, LIST, NLIST).

Az ASSEMBLER-ek általában képesek ún. relokálható (áthelyezhető) tárgyprogramok létrehozására is. Ha a tárgyprogram olyan kiegészítő információkat tartalmaz, amelyek felhasználásával a tárgyprogram egy, a fordítás során feltételezett betöltési címtől különböző tárcímen is a tárba tölthető és futtatható, vagyis áthelyezhető, akkor mondjuk, hogy relokálható.

Az ASSEMBLER-ek alkalmazásának további nagyarányú javulását eredményezi, ha az ASSEMBLER képes ún. makrók feldolgozására is. A relokálással és a makrók felhasználásával kapcsolatos részletes ismeretek tárgyalása nem célja ennek az írásnak. A téma alapos elemzése megtalálható például J. Donovan Rendszerprogramozás (Budapest, 1977. Műszaki Könyvkiadó) című könyvében.

DR. KÓNYA LÁSZLÓ

ADOK-VESZEK-CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közleteknek gépelt soronként (60 karakter) 100,- Ft, magánszemélyeknek az első sor 50,- Ft, minden további sor 20,-Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

ADOK

ATARI ST-hez felhasználói programok eladása nagy választékban. Programfejlesztést is vállalunk. Levélcím: Torday Judit, Debrecen, Tessedik S.u. 120. 4032

Brother HR-5 Mátrixprinter, RS-232 Interfész (hálózati és telepes), 3 karakterkészlet, levélminőség, elit és nyújtott betűk, gumigörgős papirtovábbítás olcsón eladó. T.: 158-154.

C16-os olcsón eladó. Irányár 8000,- Ft. Kiss István, Jákó, Zrínyi u. 2. 7525

C16 Tips-Tricks eredeti Data Becker könyv, Poke, játékprogramok, szövegszerkesztő, belső rutinok sürgősen eladók. Első érdeklődőnek engedmény! ifj. Valisko János, Budapest, Pozsonyi u. 15. 1137, Tel.: 314-817.

Commodore 16, garanciával (még nincs egy éves) + joystick interfész, sok programmal és jelentős dokumentációval eladó. Tel.: 317-033 délután két órától. Székely Tamás, Budapest, Rosenberg házaspár u. 21. 1054

Commorore 64 számítógéphez fix memóriabővítő Cartridge. Alkalmazói- és játékprogramok (monitor, assembler, copy stb.) 2 kbájttól 32 kbájtig, 649,- Fttól 1699 Ft,- ig. Turbo tape reset gombbal 699 Ft,-. Poke utasítással lapozható 2 ill. 4-szer 8 kbájtos memória. Tetszőleges memóriaterületen lévő maximum 31 kbájtos program Cartdidge-ba való elhelyezése, hozott diszkről vagy kazettáról. EPROM égetése 64 kbájtig. Tel.: 287-493 este 18-21 óráig.

Commodore 64 programok titkosítását floppyn vállalom magánszemélyeknek és közületeknek. Minden másolóval szemben biztos védelem, igen nehéz visszafejthetőség. Levélben részletesen tájékoztatom. Báldi András, Post Restant, 1364 Budapest 4.

Commodore 64-es számítógépemet eladnám kazettás egységgel, 250 kazettával, amely közel 1500 programot tartalmaz (játék, felhasználói és saját készítésű szótár, teszt, és fordítói részből áll közel 50 db). Járóka László, Budapest, Szív u. 3-5. fsz.5. 1063

C Pus/4 1551-es floppyval és Citizen 40 oszlopos kétszínű nyomtatóval eladó. Tel.: 427-646 este.

Enterprise-hoz disassembler! Teljes kódkészlettel, rendszerhívások, szegmentált és abszolút címzés. Ára: 240,- Ft. Utánvéttel elküldöm címére. Újlaki László, Budapest, Lavotta u. 24. 1104

Primo A64-es számítógépemet eladnám vagy elcserélném (Homelab 4 vagy TVC + ráfizetés). Árajánlatokat a következő címre kérem: Battancs Zsolt, Hódmezővásárhely, Damjanich u. 122. 6800

Programok C Plus/4 és C16 számítógépekre: FORTH nyelv, Turbo file kazettás fájlkezelő, Disk system BASIC és assembler fejlesztőprogram, Copy 226 lemezes/szalagos másoló, C64-C Plus/4 turbo, C64-C Plus/4 rutin és rendszerváltozó összehasonlító táblázat, Profiass C Plus/4 kétmenetes assembly fordító. A programok ára 180-600 forint, kazettán, utánvéttel, részletes kezelési utasítással. Kérésre ismertetőt küldök. Pelsőczi Gyula, Szilasliget, Ady Endre u. 36. 2145

TV-Computer (32 k) eladó, játékprogramokkal. Tel.: 111-992.

ZX-Spectrum számítógéphez kempston joystick interfész I. Irányár: 1800 Ft. Jelentkezni levélben: Kósa Péter, Budapest, Kossuth Lajos utca 83/B. 1211

ZX-Spectrum+ számítógép beépített joystick-interfésszel, joystickkal és játékprogramokkal 17 000,- Ft-ért eladó. Miskolc, Kulich Gyula u. 8. 5/3. 3529, Tel.: 69-540.

VESZEK

Nagyon sürgősen vennék ATARI 800 XL személyi számítógéphez adatrögzítő magnót. (Nem interfészt!) Lehetőleg XC-12-es típusút, de az ATARI 1010-es is megfelel. Gergely Miklós, Keszthely, Zalka Máté u. 1. 3/15. 8360

Commodore 1541-es használt, esetleg hibás drive-ot veszek. Ormos Zoltán, Budapest, 4. 1364

CSERÉLEK

C16, Plus/4 számítógépekre játékprogramokat cserélek. Raffai Gábor, Veszprém, Szabó Ervin u. 2/F. 8200

C64-es játékprogramokat cserélek. Ezenkívül cserélek még formázó és másoló programokat. Nagyon örülnék ha valaki el tudná küldeni az egész lemezmásolót. Czakó Gábor, Mátészalka, Bajcsy Zs. u. 13. II/5. 4700

Commodore 64-es számítógépre programokat cserélek (kazettán). Banyák Mihály, Mosonmagyaróvár, Széchenyi u. 19. II/8. 9200

C64-es programokat cserélek (lemezen és kazettán). Kiss Attila, Győr, Ifjúság krt. 101. III/11. 9023

Enterprise programcsere. Kálmán Albert, Veresegyháza, Kinizsi u. 50. 2112, Tel.: 143-031, 330-345

Enterprise és ZX-Spectrum felhasználói és játékprogramokat cserélek. Opovszki Sándor, Káva, Akácfa út 8. 2215

Primót és **Z80-as** fejlesztő rendszert C64-re cserélnék. Telefon: 685-140

ZX-Spectrum színvonalas játék- és felhasználói programokat cserélek. Pazsiczky Jenő, Budapest, Kút u. 5. 1024

FORDULJON HOZZÁNK BIZALOMMAL!

Jól képzett, nagy szakértelemmel rendelkező munkatársaink megoldják problémáját.

SZOLGÁLTATÁSAINK:

Vállalatirányítás fejlesztése
Rendszerelemzés — tervezés
Szoftvertermékek készítése és forgalmazása
Számítógépes szolgáltatások
Lízingelés

Szervezési és Számítástechnikai Vállalat Signal

Budapest XIII., Csata u. 8.
Levélcím: 1391 Budapest
Pf.: 207.
Telefon: 296-027
Telex:22-5736

Miért nem stupid a MUPID? avagy: Milyen az intelligens videotex-dekóder? III.

Sorozatunk előző részében a MUPID belső felépítését tekintettük át, és lehetséges üzemmódjai közül bemutattuk a legalapvetőbbet, a lekérdező üzemmódot. Most a dekóder többi üzemmódját, továbbá a vtx-szolgáltatások szempontjából kiemelkedően fontos szerkesztést és az egyéb információbeviteli lehetőségeket ismertetjük.

Mágneslemezegység üzemmód

A dekóder DISK feliratú csatlakozóján a szintén MUPID gyártmányú intelligens, Z80 alapú kettős mágneslemezegységet fogadja. A 64 kbájtos RAM-mal és 8 kbájtos EPROM-mal rendelkező központi egység két, egyenként 400 kbájt tárolókapacitásra felformázható hajlékonylemez kezelésére alkalmas meghajtót vezérel. A MUPID és a lemezegység között a SHIFT DISK billentyűk lenyomásával épül föl a kétirányú kapcsolat, melyet a lemezegység felügyel. Az adatátviteli sebesség is ezen választható ki kapcsolóval a 19 200 vagy 4800 bps értékek közül. A lemezegység saját „logo"-val jelentkezik be, és a képernyő 25. sorában a négy lemezoldal közüli választási lehetőséget kínál. Az alkalmazói felület hasonló a vtx-központéhoz, így a lemezen lévő programok és vtx-oldalak a vtx-szolgálatokban szokásos — a MUPID videotex üzemmódjánál ismertetett — protokoll szerint érhetők el.

A lemezegység beégetett programjával is alkalmas különböző lemezműveletek elvégzésére, de igazán komfortossá csak az egységhez mellékelt M(upid)-DOS programcsomag révén válik. Az M-DOS programok segítségével nagymértékben egyszerűsödik mindenféle másolás, törlés, lemezazonosítás és fájlcsoportosítás. Talán felesleges hangsúlyozni, hogy a lemezegység lehetőséget nyújt a vtx-központból lehívott információk és programok lemezre mentésére, programok helyi betöltésére és futtatására, helyi szerkesztésre (információs oldal és program is) és mindezek vtx-központba való visszatöltésére. A lemezegységgel konfigurált MUPID dekóder önálló, CP/M alatt futó HC-ként használható, de alkalmas házon belüli vtx-központ szimulálására is.

A MUPID a kezelő kívánságára felépült vtx-összeköttetés alatt is bármikor a lemezegységhez fordulhat — a telefon-összeköttetés megszakítása nélkül. Ily módon kérhetők le oldalak a vtx-központból és menthetők lemezre vagy hívhatók be lemezről olyan segédprogramok, amelyek a lekért oldal vagy program előzetes letárolása nélkül is lehetővé teszik azok további, helyi feldolgozását (vtx-oldal kinyomtatása különböző formátumokban, oldalak átszerkesztése, válaszoldal gyors kitöltése, futó programban adatlehívás vtx-adatbázisból stb.).

A lemezegység üzemmódból bármikor lehetőség van az alapállapotba — vagy élő vtx-összeköttetés esetén a vtx-üzemmódba — való visszatérésre a DISK billentyű egyszeri lenyomásával.

RAM DISK üzemmód

Ez az üzemmód csak a 128 kbájtos CMOS-RAM bővítéssel ellátott komfort MUPID változatnál lehetséges. A RAM-bővítő háttértárolóként való használatához előzőleg egy CMOS kezelő- és nyilvántartó programot kell lemezről vagy a vtx-központból a CMOS-területre betölteni. Az így kialakított RAM DISK kezelése a lemezegységéhez hasonló. A MUPID bármely üzemmódjából a kék K és a ? billentyű egymás utáni leütésével hívható, amire az mindjárt a tartalomjegyzékével jelentkezik be.

A RAM-bővítés akkumulátoros táplálású, így a MUPID bekapcsolása nélkül is körülbelül egy hónapig megőrzi a benne tárolt oldalakat, programokat.

Helyi üzemmód

Alapállapotából a MUPID a kék K és L billentyűk egymást követő lenyomásával hozható helyi üzemmódba. A kurzor a CTRL Q hatására jelenik meg. Ez az üzemmód tulajdonképpen helyi szerkesztést tesz lehetővé, a billentyűzetről — megfelelő változatszám beadásával — a teljes alfamozaikos CEPT karakterkészlet előállítható. Mivel így csak egy oldal szerkeszthető, ráadásul az is rendkívül nehézkesen, ez a szerkesztési mód csak háttértár (M-DISK, RAM-DISK) nélküli terminálkonfigurációnál jöhet számításba. A gyakorlatban a MUPID-hoz készített szerkesztőprogramokat használják, melyek a számítástechnikában járatlan alkalmazók számára is könnyen előállíthatóvá teszik a lekülönbözőbb megjelenítési szinteknek megfelelő oldalakat.

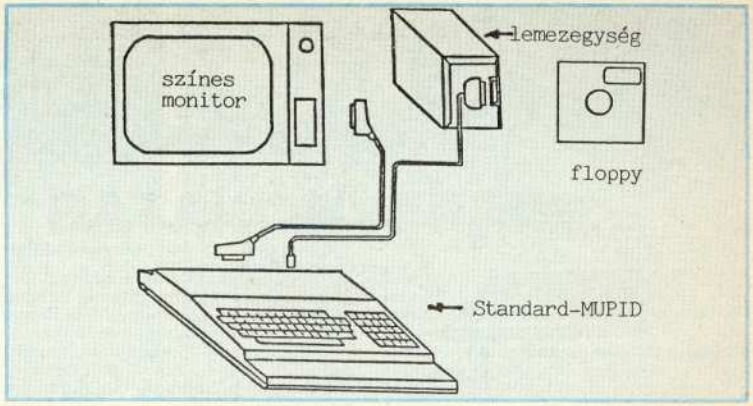
Miután áttekintettük a különböző üzemmódokat, feltétlenül szólnunk kell arról, hogy miképpen szerez tudomást a felhasználó a MUPID éppen aktuális üzemmódjáról. Erről a MUPID-hoz kapcsolt képernyő bal alsó sarkában minden üzemmódban megjelenő színes állapotmező tájékoztat. Az öt színes fényvonal közül balról az első a dekóder bekapcsolt állapotáról, a második az aktuális üzemmódról, a harmadik a külső háttértár felől érkező adatáramlás tényéről (ilyen értelmezésben a vtx-központ is így fogható föl), a negyedik a billentyűzetről a háttértár felé folyó adatbevitel tényéről, az ötödik pedig a billentyűzetpuffer túlcsordulásáról ad felvilágosítást. A harmadik kijelzés segítségével a felhasználó a MUPID-ba való betöltés esetleges befagyásáról szerezhet igen gyorsan tudomást.

Videotex-alkalmazások

Más vtx-rendszerekben alkalmazott vtx-dekóderekkel (NSZK: Raphi, Blaupunkt, Loewe; Franciaország: Minitel) összehasonlítva feltűnő, hogy a MUPID-nak nincs

külön lekérdező- vagy szerkesztődekóder változata. Komfortossága kizárólag az alapkészülékhez csatlakoztatott perifériakonfigurációtól és a betöltött programoktól függ, és természetesen az alkalmazási igény határozza meg. A felhasználó rugalmasan bővítheti a MUPID szolgáltatásait, ami anyagilag is kedvező: mikrogépcsere nélkül vezeti őt a számítástechnika aránylag magas szintű megismeréséhez. (Vigyázzunk: ez azért még mindig csak a 8 bites világ!)

A vtx-alkalmazások szempontjából kiemelkedően fontos az adatbázisokban elhelyezendő információ „videotexesítése". Ennek legkonvencionálisabb módja a vtx-oldalak dekóderrel való szerkesztése, ami történhet a vtx-központ támogatásával online módon, vagy helyileg, hardver- és szoftvertámogatással. Az előbbi a szerkesztés befejezését követően azonnal elhelyezi az adott oldalt az adatbázis megfelelő helyére, az utóbbinál általában több megszerkesztett oldal kötegelt bevitele történik távközlési vonalon vagy mágneses adathordozóról. Alább a helyi szerkesztési és egyéb információbeviteli lehetőségeket ismertetjük, melyeket az adott terminálkiépítettség és az alkalmazott szerkesztőprogram együttesen határoznak meg.



1. ábra. Egyszerű MUPID szerkesztő konfiguráció

Szerkesztés

EASY EDIT

A konfiguráció (lásd az *1. ábrát* is): a MUPID bármelyik változata, színes monitorral, továbbá M-DISK (mágneslemezegység).

A program: az M-DISK-hez mellékelt DOS 2 rendszerlemezen lévő EASY EDIT program, amely a vtx-központból is lehívható.

Alfamozaikos vtx-oldalak menüválasztással támogatott előállítására és átdolgozására szolgáló szerkesztőprogram. Előnyei:

— közvetlen karaktermegadás billentyűzetről,

— teljes mértékű támogatás a 25. sorban megjelenő menüből,

— attributumok, különleges karakterek és írásjelek, valamint mozaikkarakterek (blokk-, ferde, vonalmozaik) választása felkínált menü alapján,

— könnyű elsajátíthatóság.

Hátrányai:

— DRCS karakterek nem állíthatók elő, de DRCS-t tartalmazó oldalak átszerkeszthetők,

— nem állíthatók elő szabadon definiálható színek, de az ilyen színt tartalmazó oldalak átszerkeszthetők,

— nem állítható elő alfageometrikus oldal.

A rendelkezésre álló attributumok:

— karakterszín (8 alapszín és félintenzitású megfelelői),

— karakter-háttérszín (8 alapszín és félintenzitású megfelelői),

— képernyő-háttérszín (8 alapszín és félintenzitású megfelelői),

— aláhúzás (kezd/végez),

— karakterméret (normál, dupla széles/magas, dupla méret),

— villogtatás (normál, invertált, félintenzitású, háromfázisú előre/hátra stb.),

— karakterkészlet (ékezetek, írásjelek, különleges jelek, blokk-, ferde és vonalmozaikjelek, lásd a *2. ábrát*).

Az egyes attributumbeállítások — a képernyőszínt kivéve — csak az adott sor végéig érvényesek, a következő sor ismét az alapértelmezéssel (normál méret, villogás nélkül, fehér GO karakterkészlet-elemek) szerkeszthető, hacsak ismét nem állítunk az attributumokon.

A megszerkesztett oldal a MUPID RAM-jában kialakított négy átmeneti oldaltár valamelyikébe, mágneslemezre vagy közvetlenül a vtx-központba tölthető be. Átszerkesztésre is ezek valamelyikéből lehet oldalt lehívni.

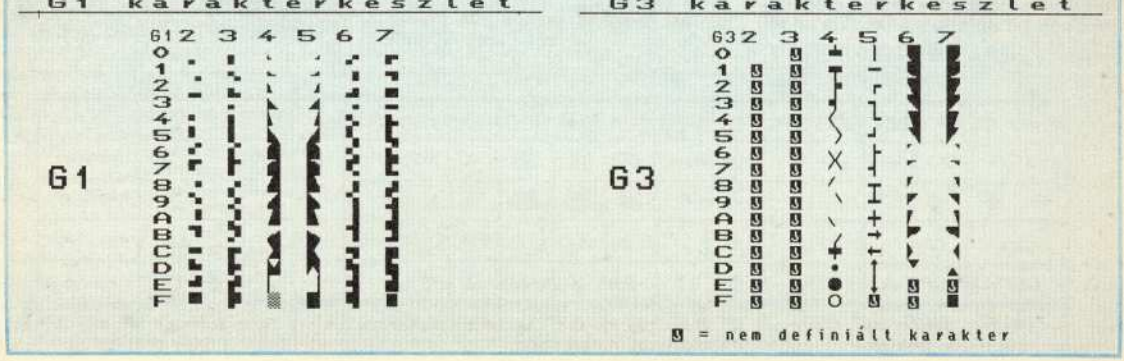
Grafikus szerkesztőprogram

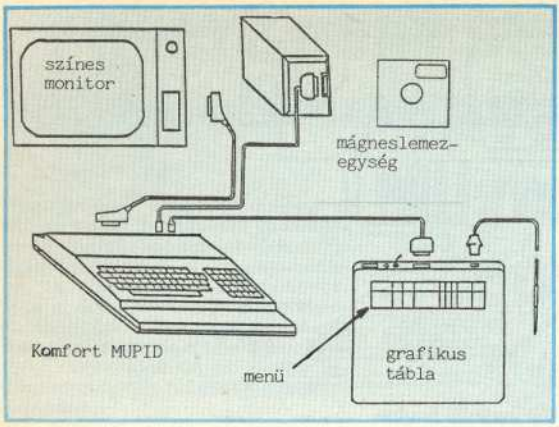
A konfiguráció (lásd az 1. ábrát is): a MUPID bármelyik változata színes monitorral, valamint az M-DISK (mágneslemezegység).

A program: az M-DISK-hez mellékelt DOS 2 rendszerlemezen lévő GRAPHIK EDITIERER program, amely a vtx-központból lehívható.

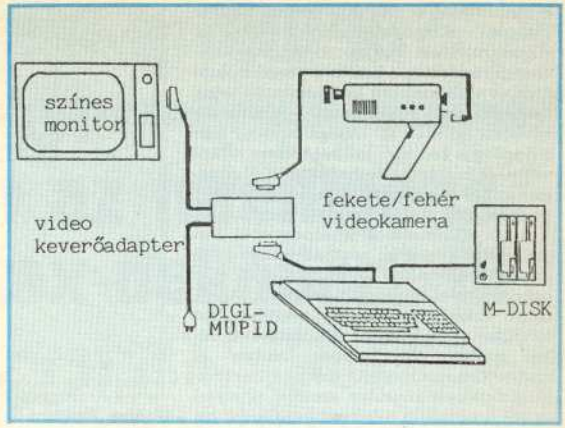
Alfamozaikos és DRCS-t tartalmazó CO megjelenítési szintű és alfageometrikus, C2 megjelenítési szintű vtx-oldalak előállítására szolgáló szerkesztőprogram. Egy oldalon a kétféle megjelenítési mód keveredhet, de az alfamozaikos felülírja az alfageometrikust. A szerkesztést kísérő funkciókat (az

2. ábra. CEPT karakterkészletek





3. ábra. Komfortos MUPID szerkesztő konfiguráció



5. ábra. MUPID képdigitalizáló konfiguráció

oldal betöltése, a szerkesztett oldal eltárolása, törlés, maszkolás stb.) itt is menüből lehet végrehajtani, de a közvetlen szerkesztés nagyrészt a CEPT karakterkészleteket és vezérlőkarakereket megadó mnemonikok (ún. Leicht Lesbaren BildCode = LLBC, azaz könnyen olvasható képkódok) felsorolásával lehetséges. A mnemonikok egy egyszerű szövegszerkesztő segítségével, a CEPT képszerkesztési szabályoknak megfelelő sorrendben írhatók egymás után egy átmeneti tárolóba. A szerkesztésre többféle lehetőség is van: a profi szerkesztők közvetlenül a mnemoniktárba írhatnak anélkül, hogy a szerkesztendő képet egyidejűleg látnák. A mnemonikok bevitelét lezárva és menüből megjelenítés üzemmódot kérve tehető láthatóvá az addig megszerkesztett kép. Lehetőség van a szerkesztendő kép és a mnemoniktár egyidejű megjelenítésére, de ez utóbbinak ekkor csak az éppen aktuális sora jelenik meg a kép legalján. Törölni csak közvetlenül a mnemoniktárból lehet. (Megjegyezzük, hogy a szerkesztő kezelése a német rövidítésekből kialakított mnemonikok miatt nehezen szokható meg.)

Ugyanennek a szerkesztőprogramnak van egy, az alfageometrikus oldalak menü általi szerkesztését lehetővé tevő, az EASY EDIT alfageometrikus megfelelőjének tekinthető programmodulja is. Itt is vannak korlátozások: például a szabadon definiálható színeket előzőleg mnemonikokkal kell definiálni, és csak azután állnak rendelkezésre. A grafikus írás attributumait (betűméret, betűtávolság, vonalvastagság, az alkalmazott karakterkészlet, a betű- és írásirány) szintén így kell megadni, s azután billentyűzetről, a menü segítségével kijelölhető helyre begépelni. A képernyőszint is a modul behívása előtt kell meghatározni.

Tekintsük át röviden az alfageometrikus megjelenítés fontosabb attributumait és a geometrikus elemek készletét. A geometrikus pontnak x, y irányú kiterjedése, többféle megjelenési formája (x, +, , o, csillag, pont) és színe lehet. A vonalszakasz, körív, csavartvonal képét vonalvastagsággal, színnel és vonalfajtával (folytonos, szaggatott, pontvonal, pontozott) variálhatjuk. A síkidomok (sokszög, körcikk, körszelet, kör, négyszög) jellemzői a kitöltési mintázat (mely lehet üres vagy tele szín, jobbra/balra sraffozott, x vagy + mintázat) és a szín.

A szerkesztőprogram az EASY EDIThez hasonló oldalletárolási/behívási lehetőségeken kívül a magnókazetta használatát is megengedi.

Komfortos grafikus szerkesztőprogram

A konfiguráció (lásd a *3. ábrát* is): a Komfort MUPID színes monitorral, az M-DISK (mágneslemezegység) és a MUPID grafikus tábla.

A program: a külön lemezen lévő KOMFORT GRAPHIK EDITIERER CMOS nevű program, amelyhez a Szerkesztői kézikönyvet mellékelik dokumentációként.

A billentyűzetről történő szerkesztés a Grafikus szerkesztőhöz képest kiegészült a csak alfamozaikos oldal képrészletét vagy egy sorát mozgató/törlő modulokkal.

Ezzel a szerkesztőprogrammal az alfageometrikus szerkesztés bővült a már megszerkesztett grafikus alakzatok eltolását, nagyítását és elforgatását lehetővé tevő modulokkal. A komfortosságot legnagyobb mértékben az ún. grafikus táblát kezelő modul növeli. A tábláról a menüvezérelt alfageometrikus szerkesztéshez hasonlóan, de annál kényelmesebben állítható elő alfageometrikus oldal. A táblára fektetett ábrát a tábla szélén levő menüben (lásd a *4. ábrát)* szereplő geometrikus elemekből és azok jellemzőit meghatározó attributumokból kell felépíteni, egy fényceruzához hasonló eszköz segítségével. A grafikus írás jellemzői is megadhatók innen, a szöveget viszont billentyűzetről kell bevinni.

Ugyancsak ezen a lemezen található egy átkonvertáló program, amely alfageometrikus oldalt DRCS karakterekből felépülő oldallá alakít. A programot az tette szükségessé, hogy a MUPID az NSZK vtx-terminálpiacán is megjelent, és az NSZK vtx-rendszerben csak DRCS-sel kiegészített alfamozaikos megjelenítést alkalmaznak. Az

4. ábra. A grafikus tábla menüje

POS	PUN		LIN	BOG	SPL	POL	SEG	SEK	KRE	REC	TEXT	VS0	VS2	N	Links oben	Mitte oben	Rechts oben	Ende
Größe	✳	X	Strich	——	-----	nicht gefüllt					TXG	TXR	TXA	o	Links Spitze	Mitte Spitze	Rechts Spitze	Eingabe
+	●	○	stärke	.....	——	ganz gefüllt						oben		r	Links Mitte	Mitte Mitte	Rechts Mitte	Farbe
LBS		CPA		MFD		—		Muster	MUD	MUF	links	Schreibrichtung	rechts	m	Links Basis	Mitte Basis	Rechts Basis	Ignorieren
LAT		CPE		MFF		Ø	1	2	3	4		unten		a	Links Unten	Mitte Unten	Rechts Unten	Beenden
WIN		VPT		CPP		5	6	7	8	9	hor	vert	frei	l	N o r m a l			

előzőek értelmében látható, hogy aránylag egyszerűen lehet alfageometrikus oldalakat létrehozni, míg a DRCS-sel való képszerkesztés — bár nagyon tetszetős, nagy felbontású grafikák hozhatók létre vele — igencsak fáradságos. A konvertálással ez a munka lényegesen egyszerűsíthető, de ennek ára a gyengébb felbontás, hiszen alfamozaiknál 480 × 240, alfageometriánál 320 × 240 a képernyőn megjeleníthető logikai képelemek száma.

Képdigitalizálás

A konfiguráció (lásd az *5. ábrát* is): DigiMUPID színes monitorral, M-DISK (mágneslemezegység), fekete-fehér videokamera, MUPID video-keverőadapter.

A program: külön lemezen lévő digitalizáló program.

A videokamerával felvett fénykép, ábra, alakzat analóg jelét a MUPID-ba épített digitalizáló modul digitalizálja és kódolással tömöríti, majd a digitális adatok háttértárba tölthetők (vtx-központ, RAMDISK, mágneslemez).

Előnyei:

— az eredetihez hű képvisszaadás,

— gyors és automatikus „videotexesítés".

Hátrányai:

— nagy a képenkénti tárigény (10—24 kbájt),

— a vtx-központból való lehívásnál igen hosszú a betöltési idő.

A hátrányokat csökkentendő, a más-más tulajdonságú képekhez különböző kódolási eljárásokat dolgoztak ki, melyek tömörítik a már digitalizált jelet. A képminőség javítására speciális digitalizálási eljárások szolgálnak, melyek az átviendő analóg videojelhez a képminőséget legjobban megőrző szürke skála digitális értékeit rendelik.

A digitális képátvitel tipikus területei: aláírás továbbítása azonosításhoz, műszaki rajzok, tervek átvitele, időjárási térképek továbbítása, körözött személyek azonosítása. (Folytatjuk)

JURENKA OSZKÁR

Fogalmak

protokoll: az információcserét vezérlő eljárás

logo: az ember—gép kapcsolatban a gépi fél. Bejelentkezése: oldal, kép vagy felirat

válaszoldal: olyan vtx-oldal, amelyet valamely információszolgáltató abból a célból hozott létre, hogy azon vtx-előfizetők információkat továbbítsanak feléje (például kérdőív-kitöltés, rendelés, üzenet, észrevétel)

attribútum: szó szerint jellemző tulajdonság; itt olyan vezérlőkarakter, amely meghatározza a megjelenítendő információ megjelenési tulajdonságait (karakterkészletet, formátumot, színt, méretet, vonalvastagságot stb.)

COMMODORE PLUS/4

DATAKONVERTER

A konverter a hosszú programlisták begépelését könnyíti meg, ugyanis ha ezzel a programmal készül a DATA-lista, akkor a soronkénti kontrollösszeggel lehetővé válik a soronkénti ellenőrzés. Ehhez mindössze egy körülbelül 10 soros BASIC töltőprogram szükséges. A töltőprogram feladata (DATA) soronként 8 egybájtos hexaszám beolvasása és összeadása, majd az összeg összehasonlítása a kétbájtos kontrollösszeggel.

A program listája is hasonló formájú, ebből pedig jól látható, hogy a BASIC-töltő először végigellenőrzi az egész DATA-listát és csak ha jó, ír bele a memóriába. Ezzel elkerülhető, hogy hibás és hiányos gépi kódú programtöredékek foglalják a memóriát.

A program elindításakor megkérdezi, hogy a konverter a memória melyik területére töltődjön, ugyanis a program relokálható. Ha nem akarunk a BASIC-területből elvenni, megadhatjuk például az 1536-os címet is, ami rendszerterület ugyan, de a program ott nem zavarja a BASIC rendszert. (Megjegyzendő, hogy igen sok TURBO használja ugyanezt a területet.)

A kontrollösszeg ellenőrzése és memóriába írása után a program még egyszer biztonságból kiírja, hogyan kell indítani a konvertálást: SYS töltési cím, kezdőcím, végcím, sorszám ahol

töltési cím:	amit elindítás után megadtunk
kezdőcím:	memóriaterület kezdőcíme
végcím:	és annak végcíme
sorszám:	az a sorszám, amelytől kezdődően a kész DATA-sorok sorszámozódni fognak. (A lépésköz egy lesz.)

Fontos tudni, hogy az új DATA-sorokat a konverter a tárban lévő *BASIC program végéhez hozzáfűzi*, tehát lehetőség van több különálló memóriaterület egymás mögé konvertálására is. Persze ehhez a BASIC-töltőt megfelelően át kell alakítani. Ugyanezen összefüggés miatt a legelső konvertálás előtt NEW-val töröljük a konverter DATA-sorait és a töltőt. (Ez a program C16, C116, C232 és Plus/4 gépeken egyformán futhat.)

NAGY TAMÁS

```
10000 DATA 20,DE,90,A5,14,85,D8,A5,0456
10001 DATA 15,85,D9,20,DE,90,A5,14,03C7
10002 DATA 85,DA,A5,15,85,DB,20,DE,0477
10003 DATA 90,38,A5,2D,E9,02,85,2D,0344
10004 DATA A5,2E,E9,00,85,2E,18,A5,032C
10005 DATA 2D,69,23,AA,85,DE,A5,2E,0399
10006 DATA 69,00,A8,85,DF,8A,C5,35,03F9
10007 DATA 98,E5,36,90,03,4C,81,86,0399
10008 DATA A0,00,84,E0,8C,10,71,8C,039D
10009 DATA 11,71,A5,DE,91,2D,C8,A5,0430
10010 DATA DF,91,2D,C9,A5,14,91,2D,03DC
10011 DATA C8,A5,15,91,2D,C8,A9,83,0434
10012 DATA 91,2D,C8,A9,20,91,2D,C8,0305
10013 DATA A9,D8,8D,9C,04,8C,12,71,03BD
10014 DATA A4,E0,20,97,04,48,18,6D,038C
10015 DATA 10,71,8D,10,71,AD,11,71,02BE
10016 DATA 69,00,8D,11,71,68,20,20,0220
10017 DATA FB,AC,12,71,91,2D,C8,8A,043A
10018 DATA 91,2D,18,90,82,90,8F,C8,034F
10019 DATA A9,2C,91,2D,C8,E6,E0,A5,04C6
10020 DATA E0,C9,00,D0,C8,AD,11,71,0479
10021 DATA 20,20,FB,91,2D,C8,8A,91,03DC
10022 DATA 2D,C8,AD,10,71,20,20,FB,035E
10023 DATA 91,2D,C8,8A,91,2D,C8,A9,043F
10024 DATA 00,91,2D,C8,18,A5,2D,69,02D9
10025 DATA 23,85,2D,A5,2E,69,00,85,0296
10026 DATA 2E,A5,D8,C5,DA,A5,D9,E5,05AD
10027 DATA D8,B0,1A,A5,D8,69,00,85,0418
10028 DATA D8,A5,D9,69,00,85,D9,E6,0583
10029 DATA 14,D0,07,E6,15,D0,03,4C,0305
10030 DATA 81,86,18,90,A0,A0,00,98,0387
10031 DATA 91,2D,C8,91,2D,C8,18,98,03BC
10032 DATA 65,2D,85,2D,A5,2E,69,00,0280
10033 DATA 85,2E,20,93,8A,4C,03,87,02C6
10034 DATA 08,00,0C,44,30,00,39,71,012A
10035 DATA 33,27,83,20,35,32,2C,33,01C3
20000 INPUT"HOVA TOLTSEM ?";BA
20001 PRINT"KONTROLL INDUL !"
20002 FORI=BTO35:C=0
20003 FORB=BTO7
20004 READA$
20005 C=C+DEC(A$)
20006 NEXTB
20007 READA$
20008 IFDEC(A$)<>CTHENPRINT"ADATHIBA A"10000+I"-EDIK SORBAN !!!":STOP
20009 NEXTI
20010 PRINT"KONTROLL KESZ !"
20011 PRINT"BEIRAS INDUL !"
20012 RESTORE
20013 FORI=BTO35
20014 FORB=BTO7
20015 READA$
20016 POKEBA+I*8+B,DEC(A$)
20017 NEXTB
20018 READA$
20019 NEXTI
20020 PRINT"OK."
20021 PRINT"INDITASA :"
20022 PRINT"SYS"BA",KEZDOCIM,VEGCIM,KEZDOSORSZAM"
20023 END
```

A MASTERFILE magyar változata

A ZX—Spectrum felhasználói programok közül az egyik legelterjedtebb, sokoldalú szolgáltatást nyújtó termék a MASTERFILE. Több változata is van forgalomban. A következőkben az igen kedvelt „ver 06" változat magyarításáról lesz szó. A módosítás eredményeként a program kezelése könnyebbé válik, és még azok is eligazodnak rajta, akiknek nincs kéznél a kezelési utasítás.

A programnak a menüből lehet utasításokat adni, de az angol nyelven megjelenő táblázatok ezt nehézkessé teszik. A módosított változat a feliratokat és táblázatokat magyarul, a gyakrabban előforduló ékezetes betűk használatával jeleníti meg, illetve nyomtatja ki *(1. ábra)*.

Az eredeti 06-os változat egy BASIC részből és az 57344-es memóriacímnél kezdődő 8192 bájt hosszú gépi kódú részből áll. A program csak a 32—127-ig terjedő kódszámú karaktereket képes kezelni, tehát a színkódokat, UDG és kulcsszavakat nem, így az ékezetes betűket a kevésbé használt szimbólumok helyére tehetjük. A ROM-ban lévő karakterkészlet szoftverrel nem módosítható, ezért a gépi kódú rész elé az *1. program* segítségével egy új karakterrészletet helyezünk el. Ez a megoldás nem csökkenti számottevően a szabad memóriakapacitást.

A program először az 56575 címtől kezdve elhelyezi az eredeti karakterkészletet, majd ezt az ékezetes betűknek megfelelően módosítja. Tehát először beírjuk az 1. programot, majd RUN-nal futtatjuk; ekkor az új karakterkészletnek megfelelően az ékezetes betűk az alábbi billentyűk lenyomásával állíthatók elő:

SYMBOL SHIFT + h = Á
SYMBOL SHIFT + x = ó
SYMBOL SHIFT + 2 = Ö

EXTEND módban:

SYMBOL SHIFT + a = á
SYMBOL SHIFT + d = ö
SYMBOL SHIFT + f = é
SYMBOL SHIFT + g = ü
SYMBOL SHIFT + u = Ü
SYMBOL SHIFT + y = É

A SYMBOL SHIFT + s billentyűk lenyomásával előálló karaktert nem célszerű módosítani, mert ezt a MASTERFILE „kényszerelválasztás" jelzésként értelmezi. Megfelelő működés esetén adjunk egy NEW utasítást, majd LOAD "" utasítással töltsük be az eredeti MASTERFILE programot; a CLEAR 57343 most nem kell, mert az 1. program erről a módosításnak megfelelően gondoskodik. Betöltés után megjelenik a főmenü. Ezután a V billentyűvel a magnóra mentés üzemmódba lépünk, majd lenyomjuk a P billentyűt, mire elindul a teljes program kimentése. Ezt BREAK-kel megszakítjuk, hogy a BASIC részhez hozzáférhessünk.

A módosításra a *2. program* való, amit célszerű előzőleg beírni és kazettára kimenteni. A program a DATA sorokat ellenőrzi, mivel a sok szám beírása közben igen könnyű téveszteni. A kimentett 2. programot MERGE "" utasítással töltsük a gépbe, majd GO TO 5-tel (nem RUN-nal!) indítsuk el. Futás után írjuk be az új karakterkészlet címét: POKE 23606, 255 és POKE 23607, 219. Ezután a MASTERFILE programot GO TO USR R utasítással kell elindítani. A program mindent magyarul ír ki, kivéve a főmenü fejlécét. Ha megfelelőek a feliratok, az előzőekben leírt módon BREAK-kel szakítsuk meg a programot, és a BASIC részből töröljük ki a most már felesleges módosító részt.

A fejléc módosítása egy kissé körülményesebb — esetleg el is hagyható. Ennek a tárban nincs fix helye, a BASIC rész után az adatfájl részeként helyezkedik el, attól függően, hogy ez mit tartalmaz. Legegyszerűbb, ha egy „üres" programot állítunk elő úgy, hogy az eddig módosított programból minden adatot, adatnevet és táblázatdefiníciót törlünk a következőképpen (csak az

1. program

```
    1 REM Karakterek
    3 CLEAR 56574
    5 FOR i=0 TO 11: READ b: POKE
 23296+i,b: NEXT i
    7 LET b=USR 23296
    8 POKE 23606,255: POKE 23607,
219
   10 DATA 33,0,61,17,255,220,1,0
,3,237,176,201
   20 RESTORE 50
   30 FOR i=0 TO 8: READ c: LET c
=c+56575
   40 FOR j=0 TO 7: READ b: POKE
c+j,b: NEXT j: NEXT i
   50 DATA 496,8,16,60,66,66,126,
66,0: REM A
   60 DATA 472,8,16,126,64,124,64
,126,0: REM É
   70 DATA 256,20,60,66,66,66,66,
60,0: REM Ö
   80 DATA 512,16,0,56,68,68,68,5
6,0: REM ó
   90 DATA 752,16,0,56,4,60,68,60
,0: REM á
  100 DATA 728,16,0,56,68,120,64,
60,0: REM é
  110 DATA 480,40,0,56,68,68,68,5
6,0: REM ö
  120 DATA 744,40,0,68,68,68,68,5
6,0: REM ü
  130 DATA 486,20,66,66,66,66,66,
60,0: REM Ü
```

1. ábra

2. ábra

```
4030 SAVE c$( TO VAL "10") LINE
VAL "4035": SAVE "SPFM"CODE VAL
"56575",VAL "8960": GO TO USR R
4035 POKE 23606,225: POKE 23607,
219: LOAD "SPMF"CODE : GO TO USR
 VAL "64951"
```

```
   1 REM Masterfile modosito
        program
   5 CLS : FOR i=1 TO 38: READ a
,b,n: LET a=a+58324: LET c=0: PR
INT AT 0,0;"DATA: ";AT 0,5;i*10
   6 FOR k=0 TO n: READ m: LET c
=c+m: POKE a+k,m: NEXT k
   7 IF c=b THEN GO TO 9
   8 IF c<>b THEN PRINT AT 0,9;"
sorban hiba !": STOP
   9 NEXT i
  10 DATA 0,493,6,45,45,45,45,99
,105,109
  20 DATA 211,2665,36,65,100,106
,32,102,105,108,101,32,110,101,1
18,101,116,255,22,1,0,255,18,1,1
7,2,16,9,32,72,73,66,94,83,32,70
,73,76,69,32
  30 DATA 391,2154,20,84,64,82,7
6,91,83,255,77,126,114,32,100,10
1,102,105,110,105,126,108,116,97
  40 DATA 461,6552,58,84,115,115
,122,97,100,97,116,98,96,108,46,
46,46,46,76,86,126,108,111,103,9
7,116,111,116,116,98,96,108,46,4
6,68,68,105,115,112,108,97,121,4
7,80,114,105,110,116,46,46,77,65
,108,97,112,109,101,110,125,104,
92,122
  50 DATA 563,1815,24,64,115,115
,122,101,103,101,32,61,32,255,23
,0,0,19,1,17,7,94,116,108,97,103
,97,32
  60 DATA 761,298,3,84,69,76,69
  70 DATA 785,2034,23,67,115,97,
107,32,97,32,70,105,108,101,46,4
6,46,46,255,83,65,86,69,32,110,1
01,118
  80 DATA 995,722,6,84,126,98,10
8,126,122,58
  90 DATA 1021,760,12,125,22,0,2
5,17,1,78,111,32,108,97,112,32
 100 DATA 1374,1236,18,109,105,1
10,100,32,18,1,83,18,0,99,115,97
,107,32,101,109
 110 DATA 1431,9186,100,75,92,11
8,101,116,107,101,122,92,32,111,
108,100,97,108,35,69,108,92,114,
101,32,49,45,57,32,115,111,114,1
16,46,66,69,108,115,92,114,101,3
2,118,105,115,115,122,97,46,46,8
0,76,121,111,109,116,97,116,98,1
14,97,46,46,46,46,46,85,69,108,1
15,92,32,115,111,114,116,32,106,
97,116,105,116,69,69,108,115,92,
32,115,111,114,116,32,116,92,114
,92,108,79,83,111,114,108,123
 120 DATA 1532,9431,102,112,116,
101,116,123,115,46,46,46,46,67,6
9,108,115,92,32,115,111,114,32,1
00,117,112,108,126,110,83,75,101
,114,101,115,92,32,125,122,101,1
09,109,96,100,46,84,64,115,115,1
22,101,103,101,47,94,116,108,97,
103,97,46,82,77,126,115,105,107,
32,116,126,96,108,126,122,97,116
,46,77,65,108,97,112,109,101,110
,125,104,92,122,46,46,46,46,81,8
6,105,115,115,122,97,46,46,46,46
,46,46
 130 DATA 2599,9238,120,78,101,1
09,32,115,122,126,109,32,45,32,9
7,100,97,116,58,18,1,83,18,0,45,
108,123,112,32,18,1,85,18,0,106,
46,46,98,229,0,0,67,66,101,116,1
25,46,46,46,46,46,46,46,46,46,46
,46,78,83,122,126,109,106,101,10
3,121,46,46,46,46,46,46,46,77,86
,105,115,115,122,97,32,109,101,1
10,125,104,92,122,46,255,69,69,1
03,121,101,110,108,92,46,46,46,4
6,46,46,46,46,85,78,101,109,32,1
01,103,121,101,110,108,92,46,46,
46,46,76,75
 140 DATA 2720,9964,119,105,115,
101,98,98,32,109,105,110,116,46,
46,46,46,71,78,97,103,121,111,98
,98,32,109,105,110,116,46,46,46,
83,83,116,114,105,110,103,115,12
2,101,108,101,116,46,46,46,77,65
,108,97,112,109,101,110,125,104,
92,122,46,46,46,46,255,84,123,11
0,121,101,122,92,32,61,255,22,18
,0,17,2,64,115,115,122,255,22,18
,0,17,2,86,126,108,32,255,22,18,
4,77,58,255,22,19,0,17,7,65,101,
116,125,32,255,22,19,0,17,7,83,1
22,126,109,32
 150 DATA 3079,1131,13,83,122,12
6,109,111,116,32,97,100,106,33,3
2,32,32
 160 DATA 3516,8719,92,84,126,11
4,111,108,116,32,97,100,97,116,9
8,96,108,46,76,83,122,92,118,101
,103,32,98,101,118,105,116,101,1
08,46,66,75,101,114,101,116,32,1
15,122,101,114,107,46,46,46,46,8
6,70,125,103,103,92,108,101,103,
46,32,118,111,110,97,108,72,86,1
05,122,115,122,105,110,116,46,32
,118,111,110,97,108,77,65,108,97
,112,109,101,110,125,104,92,122,
46,46
 170 DATA 3629,9852,100,65,100,9
7,116,111,107,32,98,101,118,105,
116,101,108,101,67,84,126,98,108
,126,122,97,116,32,99,105,109,10
1,107,46,68,68,105,115,112,108,9
7,121,47,80,114,105,110,116,46,4
6,69,84,126,98,108,126,122,97,11
6,32,115,122,101,114,107,46,76,7
0,105,106,101,32,109,97,103,110,
96,114,96,108,46,46,78,65,100,97
,116,110,123,118,32,98,101,118,1
05,116,101,108,83,75,101,114,101
,115
 180 DATA 3730,9934,105,92,32,12
5,122,101,109,109,96,100,46,73,7
3,110,118,101,114,122,32,107,101
,114,101,115,123,115,46,82,75,10
5,118,126,108,46,32,110,117,108,
108,126,122,126,115,80,66,126,10
8,46,32,97,100,97,116,116,92,114
,108,123,115,84,64,115,115,122,1
01,103,47,94,116,108,97,103,46,4
6,46,66,77,97,103,110,96,114,97,
32,107,105,109,101,110,116,46,85
,70,101,108,104,97,115,122,110,4
6,32,66,65,83,73,67
 190 DATA 4148,889,8,93,114,101,
115,32,104,101,108,121
 200 DATA 4246,415,3,83,122,92,1
16
 210 DATA 4287,988,14,83,111,114
,32,32,48,48,51,22,4,19,79,115,1
22,108
 220 DATA 4383,296,2,75,105,116
 230 DATA 4713,13604,150,75,111,
111,114,100,61,255,89,32,75,111,
111,114,100,61,255,72,111,115,11
5,122,46,61,255,83,122,123,108,1
15,61,255,77,123,108,121,115,61,
256,83,111,114,32,61,255,79,115,
122,108,111,112,61,255,80,97,112
,105,114,189,255,66,114,105,103,
104,116,32,89,47,206,73,110,118,
101,114,115,101,32,89,47,206,75,
105,116,32,89,47,206,22,2,0,17,6
,65,100,97,116,32,114,101,102,32
,84,58,255,22,6,0,17,7,83,122,12
3,108,101,115,48,48,49,22,6,11,8
3,111,114,107,92,122,48,48,49,22
,7,0,93,114,101,115,68,17,6,255,
0,0,0,22,2,0,17,6,75,101
 240 DATA 4864,10658,146,114,255
,22,5,14,77,123,108,121,115,48,4
8,48,49,255,22,4,0,17,7,88,32,75
```

```
,111,111,114,100,32,48,49,48,32,
32,72,111,115,115,122,97,32,49,5
3,48,22,5,0,89,32,75,111,111,114
,100,32,48,49,48,255,22,2,0,17,6
,86,105,122,115,122,46,116,111,1
10,97,108,255,22,2,0,17,6,70,125
,103,103,92,108,46,32,116,111,11
0,97,108,255,22,2,0,17,2,94,106,
116,97,108,126,110,255,22,5,0,17
,7,83,111,114,114,58,84,58,255,0
,22,4,9,17,7,66,111,114,100,101,
114,32,17,4,52,17,7,32,17,7,83,1
11,114,107,92,122
 250 DATA 5070,1515,15,84,126,98
,108,126,122,97,116,32,114,101,1
02,32,98,101,58
 260 DATA 5112,2136,20,78,105,11
0,99,115,32,105,108,121,101,110,
32,116,126,98,108,126,122,97,116
,111
 270 DATA 5147,16263,175,85,105,
32,116,126,98,108,126,122,97,116
,46,46,46,46,82,83,122,101,114,1
07,46,32,125,122,101,109,109,96,
100,46,77,65,108,97,112,109,101,
110,125,104,92,122,46,46,46,46,2
55,65,85,106,32,101,108,101,109,
101,116,32,98,101,97,100,46,82,6
9,108,101,109,101,116,32,106,97,
116,105,116,46,46,46,69,69,108,1
01,109,101,116,32,116,92,114,92,
108,46,46,46,78,75,92,116,101,11
6,107,101,122,92,32,101,108,101,
109,46,68,68,105,115,112,108,97,
121,47,80,114,105,110,116,46,46,
67,77,126,115,111,108,97,116,32,
101,114,114,92,108,46,46,88,84,1
26,98,108,126,122,97,116,32,116,
92,114,108,123,115,77,86,105,115
,115,122,97,32,97,32,109,101,110
,125,114,101
```

```
 280 DATA 5538,400,3,111,114,114
,61
 290 DATA 5636,3548,33,32,89,32,
104,97,32,97,107,97,114,111,237,
255,69,122,116,32,109,126,114,32
,115,122,101,114,101,112,101,108
,116,101,116,116,101
 300 DATA 5900,1818,22,89,78,84,
69,82,32,115,122,92,116,101,103,
32,49,45,49,50,56,32,98,101,116,
125
 310 DATA 5968,1171,12,65,100,97
,116,32,114,101,102,32,97,100,11
0,105
 320 DATA 6059,11060,122,85,106,
97,98,98,32,97,100,97,116,111,10
7,46,46,46,82,65,100,97,116,32,1
06,97,116,105,116,126,115,97,46,
46,69,65,100,97,116,32,116,92,11
4,108,123,115,101,46,46,46,78,75
,92,116,101,116,107,101,122,92,3
2,97,100,97,116,46,68,68,105,115
,112,108,97,121,47,80,114,105,11
0,116,46,46,71,65,100,97,116,111
,116,32,116,101,115,122,46,46,46
,46,80,71,121,111,114,115,32,97,
100,97,116,111,107,46,46,46,77,6
5,108,97,112,109,101,110,125,104
,92,122
 330 DATA 6279,465,4,109,105,32,
97,122
 340 DATA 6321,1374,13,77,97,103
,110,96,114,97,32,107,105,109,10
1,110,116
 350 DATA 6414,416,3,64,115,115,
122
 360 DATA 6425,320,2,86,126,108
 370 DATA 6435,324,2,84,126,114
 380 DATA 6833,931,10,67,115,97,
107,32,97,32,70,105,108,101
 390 POKE 63365,105
 400 PRINT AT 12,13;"OK !": STOP
```

2. program

egymás után lenyomandó billentyűket tüntetjük fel):

1. Adatok törlése: a főmenüből kiindulva —R—I—P—Y

2. Táblázatdefiníciók törlése: a főmenüből kiindulva az E billentyű lenyomásával a táblázatszerkesztés üzemmódba lépünk, majd minden táblázatdefiníciónál az —R—n—X—Y billentyűket kell lenyomni, ahol n a táblázat hivatkozási száma vagy betűje. Az összes táblázatdefiníció törlése után az M billentyűvel kell a főmenühöz visszatérni.

3. Adatnevek törlése: a főmenüből kiindulva az N billentyű lenyomása után addig alkalmazzuk az —E—Y utasítást, míg az összes piros adatnév eltűnik. Ezután az M billentyűvel álljunk vissza a főmenüre.

A már bemutatott módon szakítsuk meg a programot, és POKE utasításokkal egyenként írjuk be a megadott címekre az alábbiakat:

24014	2	24075	65
24028	32	24076	82
24029	32	24077	32
24030	77	24081	86
24031	45	24082	94
24058	1	24083	76
24065	55	24084	84
24070	32	24085	79
24071	77	24086	90
24072	65	24087	65
24073	71	24088	84
24074	89		

Az adatokat azért ne számozott BASIC sorokkal, hanem egyenként írjuk be, mert különben a fejléc abszolút címe eltolódik. A programot GO TO USR R-rel indítva, az *1. ábrán* látható fejlécnek kell megjelennie. A fejléc két oldalán a piros és kék színek is felcserélődnek!

Ezután a programot megszakítjuk, hogy a BASIC részből két sort a *2. ábra* szerint módosítsunk, vagy ha az ellenőrzést is szükségesnek tartjuk, akkor a *3. ábra* szerint — ez azonban tovább csökkenti a szabad memóriakapacitást.

Végül GO TO USR R-rel indítsuk el a programot, a V és P billentyűk lenyomásával pedig mentsük kazettára. Az új program használatánál ne felejtsük el a betöltés előtt a CLEAR 56574-et beírni! Előfordulhat, hogy a kimentett adatfájl betöltése után HIBÁS FILE felirat jelenik meg. Ilyenkor egy rövid BREAK után GO TO USR VAL „64951" utasítással újraindítva a programot, a betöltött fájl hibátlanul rendelkezésünkre áll.

VADÁSZ LÁSZLÓ

3. ábra

```
4020 SAVE c$( TO VAL "10") DATA
f$(): PRINT AT VAL "17",VAL "12"
;"VERIFY": VERIFY "" DATA f$():
GO TO USR R
4030 SAVE c$( TO VAL "10") LINE
VAL "4035": SAVE "SPMF"CODE VAL
"56575",VAL "8960": PRINT AT VAL
 "17",VAL "12";"VERIFY": VERIFY
"": VERIFY ""CODE : GO TO USR R
4035 POKE 23606,255: POKE 23607,
219: LOAD "SPMF"CODE : GO TO USR
 VAL "64951"
```

Integrált szoftver

Postázó II.

A program átírásához itt is csak azokat a megoldásokat tárgyaljuk, amelyek még nem fordultak elő. Ezért az átdolgozás előtt célszerű a sorozat korábbi részeit elolvasni.

10-es sor. A MEM változó értéke legyen mintegy 8 ezerrel kisebb, mint a BASIC által szabadon hagyott terület.

530-as, 550-es sor. Az EOF(−1) az „adatok vége" jel a szalagon. A legtöbb számítógépnél nincs ilyen függvény, ezért más megoldást kell keresni.

570-es, 7030-as sor. A CLOSE utasítás minden nyitott adatbázist lezár.

1200-as, 1300-as sor. A felső vessző a REM rövidítése ennél a gépnél.

Táblázatkezelő

Az eddigiektől eltérően most először az átíráshoz szükséges tudnivalókat közöljük.

140-es sor. Az Ä és Å a nyomtatón az [és] helyett íródott.

910-es sor. A RUN törli a változók értékét és újraindít. Egyes gépeknél nincs törlés, másoknál RUN nem lehet a programban.

1460-as, 1490-es sor. A POKE utasítás és a PEEK(függvény címe a képernyő második sorának bal oldali végét (1064) jelenti.

DR. SIMONYI ENDRE

```
10 CLS0:GOTO1700
20 N=26:DIM F¤(N,26),P(N,26),T¤(N,26),R(
12)
30 CLS:GOSUB50:GOSUB1740:FORX=1TO2000:NE
XT:I¤=CHR¤(13):S¤=" ":GOTO250
34 SOUND200,1
35 SOUND180,1:RETURN
50 F=0
60 IF CY>13 AND YP=0THENYP=13ELSEIFCY<14
ANDYP=13THENYP=0ELSE80
70 PRINTé96,"";:FOR XX=1TO12:PRINTCHR¤(9
6+XX+YP):NEXT:PRINTCHR¤(96+XX+YP);:F=1
80 IF CX>CP+2THENCP=CX-2ELSEIF CX<CP THE
NCP=CX ELSEIF F=0THEN130
82 SOUND1,1
90 PRINTé64,STRING¤(32,128);
100 FORXX=0TO2:PRINTé70+10*XX,CHR¤(XX+CP
+96);:NEXT
110 FORYY=1TO13:PRINTé65+YY*32,"";:FORXX
=0TO2:IFT¤(XX+CP,YY+YP)<>""THENPRINTLEFT
¤(T¤(XX+CP,YY+YP)+B¤,10);ELSEIFP(XX+CP,Y
Y+YP)<>0ORF¤(XX+CP,YY+YP)<>"" THENPRINTU
SINGPT¤;P(XX+CP,YY+YP);ELSEPRINTB¤;
120 NEXTXX,YY
130 PRINTéCL," ";
140 CL=(CX-CP)*10+((CY-YP)*32)+65:PRINTé
CL,">";:PRINTé32,"A";CHR¤(CX+64);CHR¤(CY
+64);"Ä ";:IFF¤(CX,CY)<>""THENPRINT"(K)"
;LEFT¤(F¤(CX,CY),23)ELSEIFP(CX,CY)<>0THE
NPRINT"(T) ";P(CX,CY)ELSEIFT¤(CX,CY)<>""
THENPRINT"(C)";LEFT¤(T¤(CX,CY),23)ELSEPR
INT
150 RETURN
160 GOSUB50:GOTO220
170 GOSUB110:GOTO210
180 GOSUB130:GOTO210
200 I¤=INKEY¤
210 PRINTé0,"":S¤="":GOSUB35
220 PRINTé1,"?"+S¤+E¤:PRINTé30,OC¤;
230 I¤=INKEY¤:IFI¤=""THEN230ELSEIFI¤=CHR
¤(13)THEN400ELSEIFI¤="!"THEN500ELSE IFI¤
="/"ANDS¤=""THENPRINTé32,SC¤:S¤="/":GOTO
220
250 IFI¤=CHR¤(92)THENGOSUB1310:GOTO220
251 IFI¤>CHR¤(31)ANDI¤<CHR¤(91)THENS¤=S¤
+I¤:GOTO220ELSEIFS¤<>""ANDI¤="Ü"THENS¤=S
¤+I¤:GOTO220
260 IFI¤=CHR¤(8)AND LEN(S¤)>0THENS¤=LEFT
¤(S¤,LEN(S¤)-1):GOTO220
270 IFI¤>CHR¤(7)ANDI¤<CHR¤(14)ORI¤=CHR¤(
94)THENIFS¤<>""THENGOSUB340:GOTO400
280 IFI¤=CHR¤(12)THENCX=1:CY=1:GOSUB50:G
OTO200
290 IF I¤=CHR¤(21)ORI¤=CHR¤(8)THENIFCX>1
THENCX=CX-1:GOTO160
300 IFI¤=CHR¤(9)ORI¤=CHR¤(93)THENIFCX<N
THENCX=CX+1:GOTO160
310 IFI¤=CHR¤(95)ORI¤=CHR¤(94)THENIFCY>1
THENCY=CY-1:GOTO160
320 IFI¤=CHR¤(91)ORI¤=CHR¤(10)THENIFCY<2
6THENCY=CY+1:GOTO160
325 GOSUB34:GOTO220
340 RETURN
400 REM
410 X=CX:Y=CY:IFLEN(S¤)>0THENC¤=LEFT¤(S¤
,1):S¤=RIGHT¤(S¤,LEN(S¤)-1)ELSE210
420 IFC¤=">"ANDLEN(S¤)=2THENX=ASC(S¤)-64
:Y=ASC(RIGHT¤(S¤,1))-64:IFX<1OR X>N OR Y
<1ORY>26THEN210ELSECX=X:CY=Y:GOSUB50:GOT
O210
430 IFC¤="/"THEN800 ELSEPRINTé0,""
440 IFC¤="é"ORC¤="."ORC¤="+"ORC¤="-"THEN
S¤=C¤+S¤:GOSUB600:F¤(CX,CY)=S¤:P(CX,CY)=
VL:PRINTé65+((CX-CP)*10)+(32*(CY-YP)),""
;:PRINTUSINGPT¤;P(CX,CY);:GOSUB140:GOTO2
10
450 IFASC(C¤)>LN AND ASC(C¤)<HN THEN S¤=
C¤+S¤:C¤="+":GOTO440
460 IFLEN(S¤)<11THENT¤(CX,CY)=S¤ELSEIF L
EN(S¤)>10AND CX<25THEN T¤(CX,CY)=LEFT¤(S
¤,10):T¤(CX+1,CY)=MID¤(S¤,11,LEN(S¤)-10)
:IFLEN(T¤(CX+1,CY))>10THENT¤(CX+2,CY)=MI
D¤(S¤,21,LEN(S¤)-20)
470 GOSUB110:GOTO210
500 REM
510 PRINTé0,"":PRINTé30,OC¤;"!";:IFOC¤="
```

```
S"THEN FORY=1TO 26:FORX=1TO N:GOTO530
520 IFOC¤="O"THEN FORX=1TO N:FORY=1TO26
530 IFF¤(X,Y)<>""THENS¤=F¤(X,Y):GOSUB600
:P(X,Y)=VL
540 NEXT
550 NEXT:GOSUB110:GOTO210
560 PRINTé32,"BEADAASI HIBA":GOSUB110:GO
TO210
600 REM
610 R=0:IFLEFT¤(S¤,1)="é"THEN700
620 FF=0:VL=0:OP¤="+":VW=0
630 IFS¤=""THEN755ELSE FOR F=1TOLEN(S¤):
W¤=MID¤(S¤,F,1):W=ASC(W¤)
640 IFW¤="+"ORW¤="-"ORW¤="/"ORW¤="*"ORW¤
="Ü"THENOP¤=W¤:GOTO750
650 IFW¤="."THENW=48
660 IFW>LA ANDW<HA THENXC=W-64:YC=ASC(MI
D¤(S¤,F+1,1))-64:F=F+1:IFYC<1ORYC>26THEN
560ELSEVW=P(XC,YC):FF=1:R(R)=F-1:R=R+1:G
OTO730
670 IFW>LN AND W<HN THEN VW=VAL(RIGHT¤(S
¤,LEN(S¤)-F+1)):W¤=STR¤(VW):F=F+LEN(W¤)-
2:GOTO730
680 IFW¤="%"THENVL=INT(VL):GOTO750ELSE I
FW¤="#"THENVL=ABS(VL)
690 GOTO750
695 PRINTé1,"HIBAAS KEEPLET":VL=1:SOUND1
,9:GOTO750
700 IFLEN(S¤)<>7THEN560ELSE R(0)=6:F=6:W
=ASC(MID¤(S¤,F,1)):IF W>LA AND W<HA THEN
710ELSE560
710 XC=W-64:YC=ASC(MID¤(S¤,F+1,1))-64:F=
F+1:IFYC<1ORYC>26THEN560ELSEVW=P(XC,YC):
FF=1

720 VL=0:IFMID¤(S¤,5,1)="O"THENFOR F=YC
TO  Y-1:VL=VL+P(X,F):NEXTF:RETURN ELSE I
FMID¤(S¤,5,1)="S"THENFOR F=XC TO X-1:VL=
VL+P(F,Y):NEXTF:RETURN ELSE560
730 IFOP¤="+"THENVL=VL+VW ELSEIFOP¤="-"T
HENVL=VL-VW ELSEIFOP¤="*"THENVL=VL*VW EL
SEIFOP¤="/"ANDVW>0THENVL=VL/VW ELSEIFOP¤
="/"ANDVW=0THENGOSUB760:GOTO730ELSEIFOP¤
="Ü"THENVL=VLÜVW
740 VW=0
750 NEXTF
755 IF FF=0THENS¤="":RETURN ELSERETURN
760 PRINTé1,"NULLAAVAL OSZTAAS":SOUND1,8
:VW=1:RETURN
800 REM
810 IFS¤="SZS"THENOC¤="S":GOTO180ELSEIFS
¤="SZO"THENOC¤="O":GOTO180
820 IFLEFT¤(S¤,1)="-"THEN T¤(CX,CY)=STRI
NG¤(10,RIGHT¤(S¤,1)):GOTO470
825 IFS¤="E"THENS¤=F¤(CX,CY):GOSUB130:GO
TO220
830 IFLEFT¤(S¤,2)="TP"THENDP=VAL(RIGHT¤(
S¤,1)):IFDP<8ANDDP>0THENPT¤=MID¤(PM¤,DP+
2,10):GOTO940 ELSEPT¤=LEFT¤(PM¤,10):GOTO
940
840 IFLEFT¤(S¤,1)="M"THENPRINTé0,"":GOTO
1400
850 IFS¤="N"THEN1200

860 PRINTé32," I-BEADAASA ESETEEN FOLYTA
TJA":GOSUB950:PRINTé32,"":A¤=INKEY¤:IFA¤
=""THENGOSUB950:GOTO860ELSE IFA¤<>"I"THE
NPRINTé33,"":GOTO210ELSEPRINTé0,""
880 IFLEFT¤(S¤,1)="U"THEN1100
890 IFS¤="BO"THEN1590ELSEIFS¤="BS"THEN16
10ELSEIFS¤="TO"THEN1600ELSEIFS¤="TS"THEN
1620
900 IFS¤="TT"THENFORX=1TO N:FORY=1TO26:P
(X,Y)=0:NEXTY,X:GOSUB110:GOTO210
910 IFS¤="T"THEN RUN
920 IFLEFT¤(S¤,1)="V"THEN1070
930 IFLEFT¤(S¤,1)="E"THEN1010
940 F=1:CP=1:GOSUB60:GOTO210
950 FORZZ=1TO99:NEXTZZ:RETURN
1000 REM
1010 GOSUB1050:GOSUB5100:OPEN"O",-1,DS¤:
O¤=CHR¤(CX+64)+CHR¤(CY+64)+RIGHT¤(STR¤(D
P),1)+OC¤:PRINT#-1,O¤:PRINTé1,"ELTAAROLA
AS";DS¤
1020 FORX=1TO N:FORY=1TO26
1029 L¤=CHR¤(X+64)+CHR¤(Y+64)
1030 IFP(X,Y)<>0THENPRINT#-1,L¤+"T"+STR¤
(P(X,Y))
1031 IFF¤(X,Y)<>""THENPRINT#-1,L¤+"K"+F¤
(X,Y)
1032 IFT¤(X,Y)<>""THENPRINT#-1,L¤+"C"+T¤
(X,Y)
1033 NEXTY,X:PRINT#-1,"EOD      "
1040 CLOSE:GOTO170
1050 S¤="":PRINTé1,"IRATCSOMONEEV";:INPU
TDS¤:IFLEN(DS¤)<1 OR LEN(DS¤)>8 THEN1050
 ELSE RETURN
1060 REM
1070 GOSUB1050:OPEN"I",-1,DS¤:INPUT#-1,I
¤:CX=ASC(LEFT¤(I¤,1))-64:CY=ASC(MID¤(I¤,
2,1))-64:DP=VAL(MID¤(I¤,3,1)):OC¤=RIGHT¤
(I¤,1):PRINTé1,"TOELTEES";DS¤
1080 IF EOF(-1)THEN1094
1090 LINEINPUT#-1,I¤:X=ASC(I¤)-64:Y=ASC(
MID¤(I¤,2,1))-64:T¤=MID¤(I¤,3,1):D¤=RIGH
T¤(I¤,LEN(I¤)-3)
1091 IFT¤="T"THENP(X,Y)=P(X,Y)+VAL(D¤):G
OTO1080ELSEIFT¤="C"THENT¤(X,Y)=D¤:GOTO10
80ELSEIFT¤="K"THENF¤(X,Y)=D¤:GOTO1080
1092 IFLEFT¤(I¤,3)="EOD"THEN1094
1093 PRINTé0,"ROSSZ IRATCSOMOO":GOTO160
1094 CLOSE:S¤="TP"+RIGHT¤(STR¤(DP),1):GO
TO830
1100 PRINTé0,"":CLS:STOP:CLS:GOSUB60:GOT
O210
1130 CLEAR9500:R=0:FF=0:VL=0:OP¤="":VW=0
:CP=4:YP=13:X=1:Y=1:CX=1:CY=1:W¤="":W=0:
S¤="":I¤="":PM¤=" #########.#########":P
T¤=LEFT¤(PM¤,10):E¤=CHR¤(128)+" ":B¤=STR
ING¤(10,32):OC¤="O":HA=91:LA=64:HN=58:LN
=47:SC¤=" - T TT TP S B V SZ N U M E"
1140 GOTO20
1200 REM
1210 PRINTé32,"":PRINTé0,"";:INPUT"BALOL
DALI OSZLOP";LC¤:LC=ASC(LC¤)-64:IF LC =>
 N OR LC<1THEN200 ELSE INPUT"JOBBOLDALI
OSZLOP";RC¤:RC=ASC(RC¤)-64:PRINTé0,"":PR
```

```
INTé32,"":IF RC > N OR RC <LC THEN200
1220 PRINTé0,"":PRINTé0,"";:INPUT"NYOMTA
TAASI KOOD";NC:GOSUB1221:GOTO1230
1221 IF SP=0 THEN PRINTé0," ":PRINTé1,"
";:INPUT"SOROS, VAGY PAARHUZAMOS (S/P)";I
¤:SP=1:IF I¤="S"THEN POKE &H3FF,1 ELSE P
OKE&H3FF,0
1222 RETURN
1230 PRINTé0,"":OPEN"O",-2,"CALCPRNT":IF
NC<>0THENPRINT#-2,CHR¤(NC)ELSEPRINT#-2,"
"
1240 FORYY=1TO26:FORXX=LC TO RC:IFT¤(XX,
YY)<>""THEN TT¤=LEFT¤(T¤(XX,YY)+B¤,10):G
OSUB8200:PRINT#-2,TT¤;ELSEIFP(XX,YY)<>0
THEN TT¤=PT¤:GOSUB8200:PRINT#-2,USINGTT¤
;P(XX,YY);ELSE TT¤=B¤:GOSUB8200:PRINT#-2
,TT¤;
1250 PRINTé0," CELLA ";CHR¤(XX+64);CHR¤(
YY+64);" KINYOMTATVA"
1260 NEXTXX:PRINT#-2
1270 NEXTYY:NC=0:PRINT#-2," ":CLOSE:GOTO
200
1300 REM
1310 GOSUB1221:PRINT#-2," ":FOR VY=0 TO
15:FOR HX=0TO31:QQ=PEEK(1024+VY*32+HX):I
F QQ>95 AND QQ<128 THEN QQ=QQ-64 ELSE IF
 QQ<32 THEN QQ=QQ+96 ELSE IF QQ>127 THEN
 QQ=46
1320 PRINT#-2,CHR¤(QQ);:PRINTé31,":";:NE
XTHX:PRINTé31," ";:PRINT#-2," ":NEXT VY:
RETURN
1400 REM
1410 FOR R=0TO12:R(R)=0:NEXTR:IFLEN(S¤)<
3THEN560
1420 S¤=RIGHT¤(S¤,LEN(S¤)-1):FOR X=1TOLE
N(S¤):S=ASC(MID¤(S¤,X,1)):IF S>LA AND S<
HA THEN NEXT X ELSEGOTO560
1430 IFLEN(S¤)<>2ANDLEN(S¤)<>4THEN560ELS
EF¤=LEFT¤(S¤,2):T¤=RIGHT¤(S¤,2)
1440 IFLEFT¤(F¤,1)=CHR¤(CX+64)THEND¤="Y"
ELSEIFRIGHT¤(F¤,1)=CHR¤(CY+64)THEND¤="X"
ELSE560
1450 S¤=F¤(CX,CY):GOSUB600:IFR(0)=0THEN5
60ELSE R=0:GOSUB140
1460 POKE(1063+R(R)),PEEK(1063+R(R))-64:
POKE(1064+R(R)),PEEK(1064+R(R))-64
1470 PRINTé0," <R>ELATIV,VAGY <N>INCS VA
ALTOZTATAAS?":I¤=INKEY¤
1480 I¤=INKEY¤:IFI¤=""THEN1480
1490 POKE(1063+R(R)),PEEK(1063+R(R))+64:
POKE(1064+R(R)),PEEK(1064+R(R))+64:PRINT
é0,""
1500 IFI¤="R"THEN1510ELSEIFI¤="N"THENR(R
)=-1ELSE1460
1510 R=R+1:IFR(R)=0THEN1520ELSE1460
1520 IFD¤="X"THENFORX=ASC(F¤)-64 TO ASC(
T¤)-64:DF=X-CX:W¤=F¤(CX,CY):R=0:Y=CY
1530 IFD¤="Y"THENFORY=ASC(RIGHT¤(F¤,1))-
64 TO ASC(RIGHT¤(T¤,1))-64:DF=Y-CY:W¤=F¤
(CX,CY):R=0:X=CX
1540 IFR(R)<1THEN1570
1550 IFD¤="X"THEN MID¤(W¤,R(R),1)=CHR¤(A
SC(MID¤(S¤,R(R),1))+X-CX):GOTO1570
1560 MID¤(W¤,R(R)+1,1)=CHR¤(ASC(MID¤(S¤,
R(R)+1,1))+Y-CY)
1570 R=R+1:IF R(R)=0THEN F¤(X,Y)=W¤:NEXT
:GOTO200
1580 IF R(R)<1THEN1570ELSE1550
1590 PRINTé31,"I";:FOR X=N-1 TO 1 STEP-1
:FOR Y=1 TO 26
1591 IF LEFT¤(F¤(X,Y),1)="é" THEN ZN=6 E
LSE ZN=1
1592 IF LEN(F¤(X,Y))=0 THEN 1597 ELSE XS
=0:FOR Z=ZN TO LEN(F¤(X,Y)):XA=ASC(MID¤(
F¤(X,Y),Z,1))
1593 IF XA > LA AND XA <HA THEN XS=1-XS:
IF XS=0 THEN 1596
1594 IF XA < CX+64 THEN 1596 ELSE XA=XA+
1
1595 MID¤(F¤(X,Y),Z,1)=CHR¤(XA)
1596 NEXTZ
1597 IF X => CX THEN P(X+1,Y)=P(X,Y):F¤(
X+1,Y)=F¤(X,Y):T¤(X+1,Y)=T¤(X,Y)
1598 NEXT Y,X
1600 FORY=1TO26:P(CX,Y)=0:F¤(CX,Y)="":T¤
(CX,Y)="":NEXTY:GOTO170
1610 PRINTé31,"I";:FOR Y=25 TO 1 STEP-1:
FOR X=1 TO N
1611 IF LEFT¤(F¤(X,Y),1)="é" THEN ZN=6 E
LSE ZN=1
1612 IF LEN(F¤(X,Y))=0 THEN1618 ELSE XS=
0:FOR Z=ZN TO LEN(F¤(X,Y)):XA=ASC(MID¤(F
¤(X,Y),Z,1))
1613 IF XA >LA AND XA <HA THEN XS=1-XS:I
F XS=1 THEN 1617
1615 IF XA <CY+64 THEN 1617 ELSE XA=XA+1
1616 MID¤(F¤(X,Y),Z,1)=CHR¤(XA)
1617 NEXT Z
1618 IF Y => CY THEN P(X,Y+1)=P(X,Y):F¤(
X,Y+1)=F¤(X,Y):T¤(X,Y+1)=T¤(X,Y)
1619 NEXT X,Y
1620 FORX=1TO26:P(X,CY)=0:F¤(X,CY)="":T¤
(X,CY)="":NEXTX:GOTO170
1700 PCLEAR1:GOTO 1130
1740 PRINTé195,"    TAABLAAZATKEZELOE
    ";:PRINTé227,"(ELECTRONIC SPREADSHE
ET)";
1750 RETURN
5100 FOR X=1 TO 6:PRINTé1," ":SOUND200,1
:PRINTé1," MAGNO-MOTOR BEKAPCSOLVA":SOUN
D200,1:NEXT X:MOTORON:FOR X=1 TO 8000:NE
XTX:PRINTé1," ":RETURN
8200 ZY¤="":FORXY=1TO LEN(TT¤):ZX¤=MID¤(
TT¤,XY,1):IF (ZX¤<>"%")AND(ZX¤<>"¤")AND(
ZX¤<>"&")AND(ZX¤<>"#") THEN8250
8210 IF ZX¤="#" THEN ZX¤=CHR¤(125):GOTO8
250
8220 IF ZX¤="%" THEN ZX¤=CHR¤(124):GOTO8
250
8230 IF ZX¤="¤" THEN ZX¤=CHR¤(96):GOTO82
50
8240 ZX¤=CHR¤(126)
8250 ZY¤=ZY¤+ZX¤:NEXT XY:TT¤=ZY¤:RETURN
9999 MOTORON:FOR X=1 TO 8000:NEXT:CSAVE"
TABLAZAT":MOTORON:FORZ=1 TO 1000:NEXT:CS
AVE"TABLAZAT"
```

IBM PC

Rejtett fájlok

Hazai számítástechnikai lapban még igen kevés szó esett az IBM PC és a vele kompatibilis gépek rejtett fájljainak használatáról. Kevesen tudják: hogyan kell a rejtett fájlokat létrehozni, és vajon a lemez másolásakor a rejtett fájlok a fájlok maradékával másolódnake?

Először a második kérdésre válaszolva: a DISKCOPY parancs az összes fájlt másolja, beleértve a rejtett fájlokat is, melyek hozzáférhetetlenek az összes többi DOS parancs számára, így a COPY számára is. Minden DOS lemez tartalmaz két fontos rejtett fájlt: az IBMBIO.COM és az IBMDOS.COM fájlokat az IBM PC gépekre, az MS—DOS esetében pedig az IO.SYS, illetve az MSDOS.SYS fájlokat. A rejtett fájlok nem jelennek meg, ha a DOS-ból kiadott DIR paranccsal vagy a BASIC-ből kiadott FILES paranccsal hívunk lemezkönyvtárt, viszont a DOS-ból kiadott CHKDSK paranccsal felderíthetők.

Illesszük be a DOS rendszerlemezt az A meghajtóba, majd gépeljük be a következő parancsot:

CHKDSK A:/V

A parancs kiadása után a lemez egy darabig forog, majd a képernyőn megjelenik a lemez összes fájljának listája, beleértve a rejtett fájlokat is. A parancs hatására megjelenő lista végén a rendszer közli, hány rejtett fájl van a lemezen. Abban az esetben, ha a felhasználó a parancsnál lehagyja a /V opciót, kizárólag egy összegzés jelenik meg, amely magában foglalja a rejtett fájlok számát.

A rejtett fájlokat többnyire a DOS rendszerfájlokban használjuk, és csak kevéssé a közönséges programozásban. Miért jó a BASIC program, ha nem használhatunk LOAD-ot, és miért jó az adatfájl, ha nem használhatunk OPEN-t az olvasáshoz? A magyarázat a legtöbb DOS parancsra is igaz: ha rejtett fájlról van szó, akkor az nem másolható a COPY paranccsal, nem listáztatható a DIR paranccsal.

A rejtett fájl létrehozása a programból nem könnyű feladat, mivel a BASIC az ilyen fájlok kezelését csak indirekt eszközökkel teszi lehetővé. A DOS-nak vannak rendszerrutinjai (megszakításai) a lemez direkt írásához és olvasásához, de az ilyen megszakításoknak a BASIC-ből való eléréséhez gépi kódú szubrutinra van szükség.

A directoryban minden fájl számára van egy ún. állapotbájt, amely azt jelzi, hogy a fájl rejtett. A DEBUG rutin használatával az állapotbájt megváltoztatható. (Az IBM PC számára ezt a rutint a DOS Supplemental Programs nevű lemez tartalmazza, a kompatibilis gépek számára pedig a DOS rendszerlemez.) Előre is figyelmeztetünk mindenkit, hogy a DEBUG egy alacsony szintű nyelven írt program, amelyhez nincs használati útmutató. Ha a felhasználó véletlenül belenyúl a lemez könyvtárába, miatt a DEBUG dolgozik, egy vagy több fájlt könnyen olvashatatlanná tehet.

```
A>DEBUG
-L DS:100 0 5 1
-D 100
0966:0100 49 42 4D 42 49 4F 20 20-43 4F
4D 27 00 00 00 00 IBMBIO COM'
0966:0110 00 00 00 00 00 00 00 60-54 07
02 00 80 12 00 00 .......'T..
0966:0120 49 42 4D 44 4F 53 20 20-43 4F
4D 27 00 00 00 00 IBMDOS COM'
0966:0130 00 00 00 00 00 00 00 60-54 07
0C 00 80 42 00 00 .......'T.B
0966:0140 43 4F 4D 4D 41 4E 44 20-43 4F
4D 20 00 00 00 00 COMMANDCOM'
0966:0150 00 00 00 00 00 00 00 60-54 07
2E 00 80 45 00 00 .......'T.E
0966:0160 41 4E 53 49 20 20 20 20-53 59
53 20 00 00 00 00 ANSI    SYS.
0966:0170 00 00 00 00 00 00 00 60-54 07
51 00 80 06 00 00 .......'T.Q
-E 10B
0966:010B 27.20
-W DS:100 0 5 1
-Q
A>
```

Aki el akarja végezni a következő gyakorlatot, az ne eredeti DOS lemezzel, hanem annak egy másolatával dolgozzon. A *listán* egy egyszerű gyakorlat látható a DEBUG használatáról. A program megcseréli az IBMBIO.COM, illetve az IO.SYS fájlokat rejtett rendszerfájlról normál fájlra. A kompatibilis gépek alkalmazásakor a fájlnevek eltérnek az itt láthatóktól: például az első bejegyzés IO.SYS néven jelentkezne.

A DEBUG utasítás beírása és az ENTER megnyomása után a gép betölti és végrehajtja a DEBUG rutint. A rutin a kötőjel (—) kiírása után várja a parancsokat. Az első DEBUG parancs (L DS:100 0 5 1) a sáv 0 tartalmát az 5-ös szektorból a lemezről a memória területére másolja át. Az „1" a parancs végén azt jelzi, hogy hány szektort kell betölteni. A parancs végrehajtása után a lemez directoryjának első részében megjelenik egy másolat a fájlról, ahol az vizsgálható, módosítható.

A D parancs begépelése esetén a DEBUG kiírja a képernyőre a következő sorokat. A képernyő középső részén minden bájt hexadecimális megfelelője jelenik meg, a jobb oldalon pedig az egyes bájtok ASCII kódjai. Közvetlenül a bájtok után az utolsó bájt a fájl állapotbájtja. Ez határozza meg a fájl tulajdonságát, és jelzi, ha a fájl rejtett.

A következő parancs, az E 10B jelzi a DEBUG-nak, hogy a felhasználó editálni kíván egy bájtot. A 10B érték azt határozza meg, hogy melyik legyen az a bájt, amelyiken változtatni akarunk. A DEBUG a bájt címével és értékével válaszol a parancsra. Ezután várja a bájt új értékének begépelését.

A W parancs az adatokat visszaírja a lemezre; a Q kilép a DEBUG-ból és visszatér a DOS rendszerbe. Ha most meg akarjuk vizsgálni a lemez tartalmát a DIR paranccsal, akkor az IBMBIO.COM vagy az IO.SYS megjelenik a normál képernyő-directoryn.

TÖRÖK PÉTER

A HCC IBM PC/XT/AT klubja újra megkezdte működését. Előadássorozatai is lesznek, A dBASE III és használata, valamint A 8088 assembler címmel - a klub hagyományainak megfelelően ingyen. Cím: BME Budapest XI., Budafoki út 8. F ép. II. lépcsőház, II. emelet 9. A foglalkozások időpontja: minden szerdán délután 3-6 óráig. Klubvezető: Bárkai János. Telefon: 664-011

Óránként húszért

„Annak a klubnak, amelyik felvenne, soha nem lennék a tagja" — mondta egyik filmjében Woody Allen. Ja kérem, ő megteheti . . . De vajon ma Magyarországon válogathat-e az, aki egy számítógépes klubba be akar lépni? És ha igen, akkor melyikre adhatja voksát? Egyáltalán: milyen manapság az ilyen klubok élete?

A kezdet

Tegyünk egy kis sétát vissza az időben. Nem sokat, csak úgy tizenkét esztendőnyit, mert akkoriban alakult meg az első számítógépes klub a világon.

A számítástechnikában az alkatrészgyártás 1975-re jutott el arra szintre, hogy a jobb módú amerikaiak már vásárolhattak alkatrészeket, s azokból maguk építhettek számítógépeket. Hiába volt azonban a jómód, mert sok vevőnek minimális számítástechnikai ismerete sem volt, így a gépépítés váratott magára. Mit tehettek hát azok, akik mégis ragaszkodtak a házilagos gépépítéshez? Összeolvastak mindenféle szakkönyvet, illetve hasonszőrű, amatőr „kollégáikkal" konzultációkat folytattak. Mindezt egy idő után klub keretében. Az első, azóta világhírűvé vált klub a HCC — Homebrew Computer Club — volt. Az alkatrészárak rohamos zuhanásával egyre nőtt a számítógépesek száma és gazdagodott a klubok köre. Ma az Egyesült Államokban mintegy háromezer klub működik, s közülük több is húszezernél nagyobb taglétszámmal büszkélkedhet.

Az amatőr gépekkel elért üzleti sikerek méltán keltették fel a világhírű számítástechnikai berendezéseket gyártók és a nagynevű kereskedők érdeklődését. Megjelent hát a piacon az óriási szériákban gyártott, különböző házi mikroszámítógépek sora, hogy végül napjainkra számos háztartás mindennapos eszközévé váljanak. Az úttörők gépépítő munkája ma már szükségtelen, viszont a vásárlók többségének most sincs számítástechnikai előképzettsége. A klubokra tehát új feladatok hárulnak: a géphasználat megtanításán túl a programok továbbadása is. Igaz, készen csaknem végtelen számú program vásárolható, de áruk cseppet sem olcsó, így a géptulajdonosok másolásra, illetve önálló programok írására kényszerülnek. Aki pedig már érti a programírás csíziójját, az nagyszerű szórakozásként újabbakat alkot.

A mikroszámítógépek terjedésének köszönhetően számítógépes klubok már nemcsak az USA-ban tevékenykednek, hanem szerte a világon, magukban hordozva a helyi adottságokat, sajátosságokat.

Hazai „vizeken"

Szerencsére Magyarország sem mentes a mikroszámítógép „fertőzésétől". Egyelőre még a háztartásokban nem nélkülözhetetlen eszköz: csak bizonyos anyagi szintet elérők engedhetik meg maguknak, hogy mikroszámítógépet vásároljanak, de a fiatalok körében mind népszerűbb a számítástechnika. Aki egyszer megismeri, az egy életre rabja lesz. Érthető hát, hogy sokan akarnának szervezett-kötetlen formában összejönni, programokat cserélni, tudásukat gyarapítani. Mindehhez kiváló keretül szolgálhatna a klub, amely adott géptípusra, különböző programokra szerveződhet. Az első 1980-ban, az amerikai „anyaklub" mintájára jött létre, és annak hozzájárulásával felvette a HCC nevet. Kezdetben itt is a gépesítés meg a számítástechnikai ismeretek oktatása volt a cél, később — ugyanúgy, mint az USA-ban — a géphasználat, a program- és információcsere foglalta le a klubtagok idejét.

A szándék kevés . . .

Az elmúlt hét évben több mint száz klub alakult. Sajnálatos módon azonban közülük több megszűnt, s ez a tény még akkor is elszomorító, ha tudjuk, hogy helyükbe újak léptek. Jószerivel a tagok is kicserélődtek: alapítókkal kevés klub dicsekedhet.

Miért a nagy mozgás a mikroszámítógépes klubok életében? Ennek több oka van. Az egyik, amitől tiszavirág életű egy klub, az, hogy a klubok alakulásakor hiányzik az összekötő cél és a domináns irányító, aki kellő elfoglaltságot, változatos feladatot adna a társaságnak. Ez szinte tüneti jelenség, és még az sem vigasztaló, ha tudjuk róla, hogy nem magyar specialitás. Az ilyen klub kezdetben megadja a tagoknak az indulásukhoz szükséges alapismereteket és különböző programokat. De amikor a tag már mindennek birtokában van, kiderül, hogy maga is elboldogul az újabb programok írásával, és nincs szüksége a klubra. Elmarad, hiszen otthon is jól eljátszhat a saját gépén. Ha azonban egy jó klubvezető keze közé kerül a társaság, akkor úgy gondolja, érdemes maradni, hiszen újabb és újabb feladatok, változatos megoldások várnak rá, és továbbképzésre is van mód. Míg tehát az első típusú klubban lezárul az ismeretszerzés, a másodikban folyamatos. Hogy a rosszra konkrét példát hozzunk: a TV-BASIC idejére számos mikroklub született, ám ahogy a tanfolyam befejeződött, már csukhatták is be kapuikat ezek a célirányos klubok.

A megszűnések másik oka — és ez úgyszintén nemzetközi jelenség —, hogy ezek a szórakozásra, tanulásra, önképzésre szerveződő társaságok permanens helygondokkal küszködnek. E klubok specialitásából adódik, hogy az eszközöket stabil helyen kellene tárolniuk, a gépek rosszul viselik a hurcolkodást. Helyiség pedig meglehetősen kevés van. De ha egy klub végre-valahára állandó teremmel dicsekedhet, még akkor sem kialthat heurékát, mert ha jól működik, annak hire megy, és mások is soraiba vágynak. A falak pedig nem nyújthatók, a gépek száma is véges, így ismét megoldhatatlan helyzetbe kerül a társaság: kinövi nehezen megteremtett kereteit.

Ami viszont már jellegzetesen magyar, az az, hogy a korábbi patrónusok, akik befogadták a klubot, a gazdasági élet változásával ellenszolgáltatásként húszforintos óradíjat kérnek valamennyi klubtagtól. Márpedig a gép, a tévé, a program manapság sem olcsó, így az óránkénti kiadás rendkívül megterhelő. És ehhez járul még egy új, nem éppen dicséretes jelenség: a programmásolásért is fizetni kell. Ez sokat elvesz a szórakozásból, és amellett törvénytelen. Mi marad annak, aki nem tudja mindezt anyagilag állni? Ismerősöktől kézen-közön másol programot, és idejét nem a klubban tölti . . .

Végül van még egy általános oka a klubok megszűnésének. Nevezetesen: bizonyos géptípusok korszerűtlenné válnak, helyükbe újak lépnek, a régiek meg mintegy elhalnak. Az úttörő géptípusok használói kikopnak az újdonságok közül.

Hogyan működhetnek klubok mégis, ilyen körülmények között? A legtöbb csak elszánt makacssággal tarthatja fent magát. Ha kidobják a társaságot az ajtón, visszakéredzkedik az ablakon, vagy tovább vándorol a következő nyitott ajtóig. Persze fehér holló azért van, olyan, amelyik helyet, irányítót, ingyen programot is talál. Meg olyan is, amelyik ma már eltér az eredeti céljától, és átment üzleti vállalkozásba.

Mi a megoldás? Az elképzelésekről szívesen adnánk közre az olvasók írásait is. Mi mindenesetre nagyobb állami támogatást várnánk mind anyagiakban, mind erkölcsi elismerésben. E kettő ugyanis még várat magára.

KRASZNAI ÉVA

OLVASTUNK..

Updike regényeiből, elbeszéléseiből eddig is érezni véltük, hogy az író számára nincs „két kultúra", nem válik szét a természettudományos és a humán műveltség. Nem véletlen, hogy egy ilyen regényt éppen a számítógépek szülőföldjén írtak meg, ott, ahol a jobb (az ún. „ivy league"-be tartozó patinás) egyetemek humán fakultásain is általában kötelező legalább egy természettudományos tárgy felvétele (enélkül a középkori angol irodalom tanára sem nagyon lehet valaki!).

A cselekmény egyik szála szerint a 28 éves Dale Kohler Isten létét — mint tényt — szeretné számítógéppel bizonyítani, és támogatásért fordul az egyetem humán fakultásának teológiaprofesszorához, a középkorú Roger Lamberthez. Kettőjük találkozásainak és párbeszédeinek feszültségét fokozza, hogy a lelkészből teológiaprofesszorrá lett Roger ironikus távolságtartással figyeli a rajongó, megszállott számítástechnikus kétségbeesett és szerinte nemcsak reménytelen, de kissé eretnek erőfeszítéseit. Dale végül is Roger taktikus segítségével megkapja az áhított támogatást a projektjéhez, úgy, hogy a gépidőt a természettudományos fakultástól lopja hozzá (fizessenek a „gazdagok!").

Dale hivatalos munkaidejében számítógépes grafikával — animációval — foglalkozik. A számítóközpont kiemelt kutatási területe a mesterséges intelligencia. Százával kapcsolnak itt össze minikszámítógépeket, hogy egy-egy nagyobb keresési problémát modularizálni tudjanak. Módszereket keresnek arra, hogyan lehet a keresőfa exponenciális növekedésének gátat szabni.

Az alábbiakban röviden bemutatjuk azt a részletet, amely Dale munkahelyét — a számítóközpontot — írja le, így:

„Április első péntekén estefelé Dale egy zsírfoltos zacskóba dugott, kihülőfélben lévő melegszendviccsel, egy kis kartondoboz zsíros tejjel és néhány, fóliába csomagolt törött süteménnyel az 1978-ban emelt épület felé baktat. Az épület az egyetem erőszakosan bővülő számítástechnikai kutatási-fejlesztési létesítményeinek ad helyet. A betonkocka — mindegyik oldalán kilencszer kilenc ablakkal — fenyegetően magasodik a környék ütött-kopott, halálraítélt bérházainak sorai fölé. Az összes ingatlan az egyetem tulajdona és várja sorsát: az egyetem terjeszkedését. Az életnek kevés a nyoma a nagy épület árnyékában. Az egyformán mélyen ülő, homokszürke keretes, ferdén bevágott ablakok olyanok, mint lőrések egy golyóbiztos bunkeren. Ezen az estén a türkizkék égen kisebb, idegesebb fehér foszlányok váltották fel a tél egyhangú felhőkabátját. Egyszerre minden moccanna — a fák ágainak hegyén az új zöld, a sár a talp alatt, minden más szeretne lenni, mint ami. Dale torkában érzi a gyomrát, tartalma marja belülről — a bűntudattól. Elfogadta a pénzt a projektjére, nagyobb falatot harapott, mint amekkorát le tud nyelni.

Bár a normális világ munkaideje már lejárt, és a Kocka hatalmas rózsaszínű márvány előcsarnokában sincs senki a Dale rétegezett belépőjét ellenőrző egy szem darvadozó biztonsági őrön kívül, az igazi alkotómunka csak most kezdődik, miután az övénél biztosabb profitot hozó projekteknek átengedett nappali munkának már vége. Dale beszáll az egyik lőporkék színű liftbe, és benyomja a 6-os gombot.

A Kocka földszintjét a recepció, a public-relations munkatársainak irodái és a műszaki könyvtár foglalja el két részlegével — a számítógép-tudományival és a nagy programozási nyelvek (LISP, FORTRAN, PL/1, Pascal, ALGOL, ennek elődje a Plankalkül és utódja a JOVIAL) részlegével —, meg itt kapott helyet egy kicsi, szórakoztató múzeum is. Láthatók itt abakuszok, inka quipusok, egy tizenhetedik századbeli logarléc, Pascal fogazott, kilincsműves kerekeinek és Leibnitz léptető szerkezetének rajzai, Charles Babbage korszakos Analytical Engine-jének falméretűre kinagyított jellegzetes részletei a Jaquard-kártyákkal és a számjegyek tárolására szolgáló kerekeinek ezreivel, válogatott lapok Lovelance matematikai jegyzetfüzeteiből, továbbá egy ugyancsak tőle származó hímzett vászonzsebkendő, az USA 1880. évi népszámlálásánál használt Hollertith-lyukkártyák mintapéldányai, a Harvardon 1944-ben üzembe helyezett Automatic Sequence Controlled Calculator fontos darabjai, egy szétszerelt akkumulátor az ENIAC-ból (a világ első, valódi elektronikus számítógépéből, melyet Philadelphiában terveztek furcsa, ódivatú bombák és lövedékek röppályájának számításához), amely tíz — egyenként tíz elektroncsövet tartalmazó — gyűrűs számlálóból áll."

És itt szakítsuk meg az idézetet, álljunk meg egy pillanatra. Miért támad az embernek az az érzése, hogy más, Updike-nál sokkal kevésbé jelentős szerzők a kis múzeum találó és szakszerű leírása helyett beérték volna például ilyesféle, nyegleséggel határos, tudatlansággal hivalkodó megfogalmazással: . . . itt található egy múzeum is, mely tele volt mindenféle bonyolult és érthetetlen szerkezettel . . . Minden krimiben találkozhatunk hasonló olcsó megoldással.

Gondoljunk bele: Updike regényében nem a számítástechnikai szakember (Dale) látja így a dolgokat. Mint a könyv címe is utal rá, ez a történet Roger-féle verziója. A teológus, a nem szakember képzeli el így a történetnek azt a részét, amelynek nem lehet tanúja. Mindenesetre biztosak lehetünk abban, hogy így csak az az írózseni képes leírni egy ilyen múzeumot, aki valamennyire tisztában van a számítógépek történetével, akinek számára ez a történet az egyetemes kultúra része.

Folytassuk most a Kocka bemutatását Updike-kal: „. . . az első és a második emeleten az adminisztráció irodái vannak tanácstermekkel és egy csupa-rozsdamentesacél-konyhával, ahol valami déli harapnivalót és tekintélyesebb látogatók számára laktatóbb előételeket lehet összeütni. A Kocka dolgozóinak épülésére egy tornaterem is van itt (súlyzógéppel és a balkonon egy futógéppel), továbbá meditációs szoba (gyékényekkel), egy háromágyas gyengélkedő, meg kerékpár- és moped-tároló — ezeket be kell hozni az épületbe, különben ellopnák őket.

A harmadik emeletre szóló gomb a legtöbb liftben le van ragasztva, és ahol nincs, a gomb ott is hatástalan, hacsak egy külön billentyűzeten nem adják meg mellé a hetente cserélt számjegykódot is. A harmadik emeleten folyó munka titkos, de ebből az említeni-sem-szabad munkából származik az a pénz, amelyen az egész Kocka nyugszik. Azok, akik a harmadik emeleten dolgoznak, ezt sohasem vallanák be magukról, mégis, fel lehet ismerni őket a többiekhez viszonyítva formális öltözékükről — öltönyről, nyakkendőről —, miközben még a kutatás és fejlesztés fő-fő főnöke, egy Benedetto Ferrari nevezetű, folyton lelkesedő olasz-amerikai is magasnyakú pulóverben vagy kigombolt nyakú selyemingben jár, mely látni engedi vastag aranyláncát vagy a már illatát vesztett kis cédrusfa-mütyürt a nyakán. Valaha nagy matematikus volt, jó ér-

és olvasmányaink közül most kivételesen egy regényt emelünk ki (John Updike: Roger's Version, Fawcett Crest, New York, 1987.), melynek az az érdekessége, hogy a számítástechnika nem csupán semleges, mással helyettesíthető, korfestő díszlet benne, hanem lényeges, meghatározó közeg, a mondanivaló fontos részének hordozója.

zékkel az elegáns egyszerűsítésekhez. Ferrari elkápráztatja az igazgatósági tagokat, telefonon egyenesen elbűvöli azokat az elgyötört washingtoni embereket, akiknek az a feladatuk, hogy — akárcsak a régi idők szénrakodói — kilapátolják napi kvótájukat az elfogyni nem akaró nemzeti vagyonból.

A negyedik emelet elsősorban Ferrari kedvenc projektjének színtere: itt az emberi agyhoz hasonló szilícium-hardvert fejlesztenek mesterséges intelligencia céljaira, habár ha azt nézzük, mennyi hasznot hoztak eddig az emberiségnek a már meglévő különféle balsorsú intelligenciák, egy még újabb, mechanikus működésű intelligencia fabrikálásának célja nem egészen olyan világos, mint az a bátorító vigyor, melyet a főnök minden oldalról begyűjthet, amikor kedvenc részlegét látogatja. Boldogsága talán leginkább Pygmalioné vagy dr. Frankensteiné mindazok közül, akik eddig már megkíséreltek elbitorolni az isteni jogot, hogy életet leheljenek az agyagba.

Az ötödik emeleten van az egész ház lelke — itt állnak a CPU-k tömött sorai: a VAX 785-ösök, a Symbolics 3600 LISP gépek, a Kocka saját tervezésű MU jelű gépe. Darálják, zabálják a számításokat a nap huszonnégy óráján keresztül, zúgó ventilátorok óvják őket a túlmelegedéstől, felszedhető álpadló védi és egyben hozzáférhetővé is teszi az őket összekötő mérföldnyi hosszúságú idegpályákhoz hasonlatos kábeleket, melyeken bájtok milliárdjai futnak nemcsak a szomszédos emeleteken elhelyezett megjelenítő-feldolgozó egységekhez, hanem — modemeken, mesterséges holdakon keresztül — olyan stratégiailag fontos helyeken működőkhöz is, mint Palo Alto, Nyugat-Berlin és Izrael. Dale, hogy lehűtse forrongó fejét, szeret néha fel-le sétálgatni itt a remegő padlókon, a dobozba zárt áramkörök és felpolcozott mágnesszalagok között, e csaknem szellemi működés hatalmas zúgásában, melyben valahogy mégis ott van a tengeri hajók gépházának otthonos csöpögése és vibrálása is, mely csak a kábelekkel és kézi csatlakoztatásokkal veszkölődő, szutykos kezű gépészek bátorító káromkodásaival együtt teljes.

A hatodik és a hetedik emelet kuckói a Kocka kevésbé kényeztetett munkatársai számára vannak fenntartva, a nyolcadikon van a légkondicionáló berendezés. Itt vakablakok vannak, kielégítendő az építész posztmodern igényét az őszintétlenségre és az üres szimmetriára.

Dale kiszáll a hatodik emeleten. Itt van a kávézó is, mely ötkor bezár. Itt vannak a hallban a meglehetősen elnyűtt gépek, melyekből érmék ellenében minden elképzelhető időben kávé, tea, húsleves (csirkéből vagy marhából), cukorka, sült krumpli, dobozos üdítőital és még háromszögletű csomagolt szendvics is nyerhető megfelelő számjegykóddal kiválaszthatóan. A számítógép-forradalom közkatonái: nagy, ütött-kopott dobozok, melyek általában makacs megbízhatósággal működnek, bár időnként hirtelen üzemzavar-kitöréssel fellázadnak: a kávé megállíthatatlanul ömleni kezd belőlük, kigyullad rajtuk a piros felirat: KIFOGYOTT, miközben a zacskó tele pirított kukoricapehellyel ott látható teljes nagyságában a műanyag ablak mögött.

A hatodik emelet egyben a hulladék birodalma is, a papírpoharaké, az eldobott szalvétáké, az egymásra tűzött hirdetéseké, melyek úgy néznek ki, mint az ablakok egy raszterdisplay-n, csak nem lehet őket gombonyomással mozgatni, hanem körömmel kell kihúzni és hüvelykujjal visszanyomni hozzá a rajzszögeket. Ezeknek a hatodik emeleti kompjuter-mágusoknak a táblái, ajtói tele vannak ragasztva a közismert rajzfilmek atavisztikus állatfiguráival . . . Mintha ezek a fiatalok agyuk gyors működéséért érzelmi elmaradottsággal fizettek volna."

Aki ezeket a sorokat így leírta, az jól ismeri a számítástechnika világát, akár Losonczi Ágnes az orvoslását. (A kiszolgáltatottság anatómiája, Magvető, 1986.) A kép hiteles. Egyáltalán nem hat benne mesterkélten, hogy a cédulákkal összevissza tűzdelt hirdetőtáblát Updike egy multi-window-s raszterdisplay-hez hasonlítja (és nem fordítva: nem az ablak-technikát magyarázza a hirdetőtábla-metaforával!). Nem kell Amerikába menni, hogy a „hatodik emelet" világát megtaláljuk, ilyen az Budapesten, Új-Delhiben, Varsóban is, legföljebb az állatfigurák mások, de azok se nagyon.

Eddig a környezetről volt szó, most a munka tartalmi részéhez közelít az író, Updike. Figyeljük meg, hogyan:

„Dale munkatársai közül csak kevesen vannak benn ezen a közbenső órán: a tavasz és a tanítási szünet sokakat hazaszólított. Allston Valentine, egy robotikával foglalkozó ausztrál látható két ajtón keresztül, mintha ő is be lenne vágva a képernyőre a szétszerelt, sokkönyökű emelőkar mellé, melynek erőhatásábrája vektordiagram formájában türelmesen ott világít a display-n. Isaac Spiegel, aki az MIT-n töltött kezdő évei óta a számítógépes fordítás elérhetetlenül mély struktúráival küzd, szótárakkal, nyelvtankönyvekkel és ügyetlen agancsként szétágazó Chomsky-diagramokkal telezsúfolt kuckójában egy doboz Michelob-sör mellett ül. A beszéd, a nyelv, mely olyan természetesen ömlik, fröcsög minden szájból, akár a nyál, ellenállóbbnak bizonyul az elemzéssel szemben, mint az enzimek. Spiegel lassan belekopaszodik szakmája szolgálatába: mindenütt szőrös, kivéve a fejebúbján, ott egy sapkányi kopasz folt díszeleg.

. . . Dale saját kuckójába megy, melyet egy szőke, szánalmasan laposkeblű továbbképzéssel, Amy Eubankkal oszt meg. A számítógépes grafikai projekt, melyen a lány dolgozik, az alakfelismerés kvantitatív megközelítésének titkait kutatja, a madarak és rovarok specifikus jegyeitől az emberek bizarr jellegzetességéig, mely lehetővé teszi, hogy barátok, családtagok akár olyan nagy távolságról is felismerjék egymást, ahonnan már az összes számszerűsíthető ismertetőjegy és értékelhető arány elmosódik. Képesek vagyunk ruhába bugyolált ismerőseinket hátulról egy háztömbnyi távolságról is azonosítani. Hogyan? Dale megtudta Amytól, hogy a rovarok a spectrum ultraibolya tartományában messzebbre látnak, mint mi, emberek, és a virágokon ennek megfelelően eligazító jelek vannak, melyek a nektárról adnak tudósítást: olyanok ezek, mint a lepkék szárnyán az udvarlásra utaló jegyek. Egész angyali párbeszédözön zajlik körülöttünk, és mi nem látjuk. Ez a felismerés érthetetlenül zavarja Dale-t — mert persze ott vannak a nyelvek is, melyeket Dale nem beszél, meg a keresztényi hit alaptétele, hogy van a tudásnak egy felettünk álló birodalma is, és hogy Isten útjai nem mindig a mieinkek is. A kutyák inkább szagolják, hallják a világot, a költöző madarak valahogyan képesek a Föld mágneses erővona-

lainak olvasására: mégis, az a gondolat, hogy a virágok olyan mintázatokat hordoznak, amelyeket csak a rovarok képesek felismerni, sérti Dale-t. A szem a lélek ablaka, atavisztikusan bízunk abban, hogy a rajta keresztül nyert információ teljes. Percipi est esse."

Mesterséges intelligencia, robotika, beszédértés, alakfelismerés foglalkoztatja tehát Dale környezetét. Ahogy közeledünk Dale kuckójához, egyre részletesebb expozícióját kapjuk a vizsgált problémáknak. Tényleg: mitől „Józsi bácsi" Józsi bácsi? És ha összeráncolja a homlokát? Bennünket ugyan nem téveszt meg, még azzal sem, ha Mikulásnak öltözik. És egy kézzel írt A betű ezer és ezer eltérő alakzatban mégis mitől A? És már mitől nem az? A szövegkörnyezettől függetlenül is képesek vagyunk leírójának szándékát felismerni (már azt a szándékát, hogy éppen oda egy A betűt akart írni). Igen fontos, nagy gyakorlati jelentőségű kérdés ez. Ha megoldjuk (most kezdenek piacra kerülni az első szövegolvasó perifériák!), akkor például lexikonokat egyszerűen be lehet olvastatni a számítógépekbe, nem kell „begépelni" őket (ha valaha is vállalkozna valaki ilyen nagy feladatra).

Következzék most Dale munkája a könyvből:

„Amy — akárcsak Dale az animációs grafikához — a négyszázezer dolláros Venust, a VAX 8600-ast kell hogy használja. Négyórás váltásokban dolgoznak, hogy mindkettőjüknek kizárólagos hozzáférése lehessen a géphez, ezért ritkán vannak együtt a kuckóban . . . (Dale) beüti a log-in nevét és az azonosítóját, majd hívja programját — a DEUS-t. Ujjaival felvarázsolja a képernyőre a transzformációs menüt, melynek mindegyik tételéhez a képernyő bal szélén egy kis szimbólum és egy kis ablakocska tartozik, amelybe kormányozható a fénylő, háromszög alakú kurzor a jobb keze ügyébe eső, elektro-optikai eleven működő egérrel. Még egy mondat a billentyűzeten, és a képernyőn gyorsan, de nem kivehetetlenül gördülni kezdenek az objektumok: FA, KAROSSZÉK, VÍZIBOLHA, SZÉNMOLEKULA, melyet ő maga vagy valamelyik számítógép-grafikás diák vitt be a gépbe huzalvázas modell formájában vektorról vektorra, szögről szögre haladva. Egyesek közülük tiszta sokszöghálók, míg mások olyan háromdimenziós felületek, amelyeket polinomiális egyenletek segítségével ragasztottak össze, és amelyek leképezése a kétdimenziós térre még a CPU óceánnyi kapacitásához képest is igen nagy mennyiségű számítást igényel. Az alkalmazandó térelem matematikailag teljesen specifikált ábrázolásának megfelelően az óriás kapacitású RAM-ok kapcsolóinak nyitott vagy zárt, elektromos zsebeinek teli vagy üres állapotaiban tükröződő 0-sok és 1-esek hosszú sorát kell minden egyes esetben elhelyezni a tár megfelelő helyén. Ezekhez Dale a billentyűzeten keresztül, a processzornak adott parancsok útján fér hozzá.

A világ így — stilizált formában — Dale ujjaiban létezik. Áhítat vagy még inkább félelem hatja át, keze elbizonytalanodik. Nincs határozott szándéka, nincs feldolgozó programja a fő program prométheuszi címében ígért végeredmény előállításához . . . Bemelegítésül Dale ráállítja a nyughatatlan-szolgálatkészen reagáló fényes, háromszög alakú pointert a SZÉNMOLEKULÁ-ra, és az ábranagyságot $10.0 \times 10.0 \times 10.0$-esre szabályozva forgatni kezdi a képernyő x tengelye körül $x = 100$ érték mellett, így:

```
(ROTATE
 (MOLECULE        (PROTEIN 293))
 (ANGLES
      (FROM ALPHA)
      (TO DELTA)
      (STEPS (*        0.001        (-DELTA ALPHA))))
 (SHADE        S3)
```

Lassan, minden harmincad másodpercben újra számítva forogni kezd a soklábú fénylő molekula az y tengely láthatatlan fonalán. Dale kegyetlen és perspektivikus ábrázolást kér, a nézőpontot közelre állítva, melynek hatására a számítások a gyorsan, táblázatosan közelített cosinusok és sinusok fáradságosan darált ciklusain keresztül lassanként legyőzik a képfelfrissítési időt, és rángva, érezhető erőfeszítéssel létrehozzák a mozgó vektorokat: a pókszerű végtagok recsegve-ropogva mozdulnak, a szenet alkotó atomok, amelyeket a csúcspontok ábrázolnak, olyanok a békésen szürke képernyőn, akár a csillagok . . ."

Térjünk vissza az idézett szöveg egy mondatához: „A világ így — stilizált formában — Dale ujjaiban létezik." Tömör és találó megfogalmazása ez annak a gondolatnak, hogy a számítógépeinkkel dolgozva a feladatok világát szimbólumok, fizikai mintázatok formájában tükröztetnünk kell a gépben: a világot le kell fordítanunk a gép számára. Ennek a folyamatnak egy része Dale fejében zajlik, itt alakul át karakterek bebillentyűzendő sorává. A világ így valóban — kódolva — Dale ujjaiban „létezik".

Nagyon sok minden van „belevarázsolva" ebbe a kis részbe. E rovat írója is félve nyúlt a fordításhoz, és reméli, egyszer egy nála avatottabb irodalmár pontosabb és ihletettebb interpretációjában is módja lesz a szöveget magyarul újra olvasni. Az Updike-művek magyar kiadója — az Európa — talán dolgozik is már a mű magyar változatán. Reméljük, a Mikromagazin e cikkével neki is tudtunk segíteni valamit a szakmai részek, a szakmai kifejezések fordításában.

Hosszú oldalakon folytathatnánk a könyv szakszerűen poétikus (helyesebben: szakszerű és poétikus — ha nem lenne szakszerű, poétikus sem lehetne) részeinek bemutatását. Nem tesszük. Reméljük, a könyv előbb-utóbb megjelenik magyarul is. Azzal búcsúzunk, hogy az innováció, a felzárkózás egyik fontos feltétele, hogy a műszaki/természettudományos kultúrát ne vegye körül a társadalom nyegle és saját érdekei ellen való idegenkedéssel, hanem integrálja, humanizálja azt. E vonatkozásban is üzenet lehet számunkra Updike legújabb könyve.

— KE —

64 k-s memóriabővítés beépítése C16-ba és C116-ba = EL-KA
- A legjobb, mert átkapcsolható 16 k-64 k között és a gép fogyasztása csak kb. 2%-kal nő!
- A legbiztosabb, mert 1 év garanciát adunk a bővítésre!
- A leggyorsabb, mert 24 órán belül elkészül (vidékieknek még aznap)!
- A legolcsóbb, mert csak 2600 Ft-ba kerül beépítéssel és átkapcsolóval!

EL-KA GM.: 1141 Budapest, Lipótvár u. 59.
Telefon : 346-933 (délután)

µINFORM

A tartalomleírások az alábbi folyóiratokban megjelent programlistákról készültek:

A folyóirat neve	Kódja
64'er Magazin	64er
Chip Magazin	chip
Commodore Horizons	coho
Commodore Microcomputers	comi
Compute!	cute
Computer Persönlich	pers
Happy Computer	happ
hc - Mein Home-Computer	hc
mc - Zeitschrift	mc
Run /USA/	run
Sinclair User	sinc
Your Sinclair	ysin

A tartalomleíró szóláncokat permutáltuk, a szóláncvariánsokat pedig alfabetikusan rendeztük.

A tartalomleírás egy szóláncból áll, majd a listában ezt követi a forrás megjelölése a folyóirat azonosítójával, a megjelenés dátumával és a cikk előkeréséhez a kezdő oldalszám és a terjedelem megadásával. A mellékelt lista értelmezéséhez még az alábbiakat kell tudni. A tartalomleírás szóláncában elsőként a téma átfogó megnevezése, utána a számítógéptípus(ok), ezt követően a szűkebben jelölt tartalom meghatározása szerepel, majd esetlegesen néhány, a közleményt minősítő adat (például : cikksorozat).

A forráshely karaktersorozatát nyíl vezeti be, melyet a / jelig a folyóiratok azonosítója, a két / jel között az évszám, folyóiratszám és kötőjellel a kezdő oldalszám követi, a végén pedig a közlemény teljes oldalterjedelme áll.

A folyóiratok a SZÁMALK szakkönyvtáraban (Budapest,XI.,Szakasits Á. út 68. Nyitva: 8-tól fél 5-ig. Tel.: 853-111/251) is föllelhetők. A kiválasztott anyagról másolat rendelhető az alábbi formában:

SZÁMALK Szakkönyvtára
Budapest, 112. Pf.: 146. 1502

Megrendelem a Mikroszámítógép Magazin 1987/ sz. alapján a következő folyóirat-oldal-másolatokat:

Kód: _____ Példányszám: _____
Kód: _____ Példányszám: _____
Kód: _____ Példányszám: _____

A megrendeléshez csatolom az oldalankénti 8,- Ft-os szolgáltatási díj befizetését igazoló csekkszelvényt.
Dátum, név, pontos cím.

PROGRAMLISTA
adatbazis kezeles commodore 64 kotetlen adatstruktura tomoritett tarolas a ramban -\run2 '86.10-121 '6
PROGRAMLISTA
adatrendezes commodore 64 lebegopontos szamtomboK gyorsrendezese select segedtomb alkalmazasa
-\run2 '86.10-87 '2
PROGRAMLISTA
adatrendezes commodore 64 szam/karakter valtozotomboket kezelo parancsoK autolagos betoltes
-\run2 '86.10-113 '2
PROGRAMLISTA
assembler programozas commodore 64 makro-assembler monitorral C6510 package -\run2 '86.12-107 '13
PROGRAMLISTA
atari st 68kPkepeK integralasa assembler programokba assemblersorgeneralas -\happ '86.12-129 '2
PROGRAMLISTA
atari xl/xe nano dos gepikodu fileoK gyors betoltese
-\happ '86.12-125 '3
PROGRAMLISTA
barkacsolas cikksorozat eprom beegeto -\run2 '86.11-24 '2
PROGRAMLISTA
basic programozas commodore 64 1 kbyte-os utasitaskeszlet-bovito
-\happ '86.12-60 '4
PROGRAMLISTA
basic programozas commodore 64 50 uj utasitas strukturalt programozashoz
-\run2 '86.11-112 '13
PROGRAMLISTA
cikksorozat commodore 16 grafika korellipszis rajzolas es nyomtatas
-\run2 '86.10-48 '4

PROGRAMLISTA
commodore 128 lemezegyseg 1571 boot rutin kezeles autoload valtozatoK
-\run2 '86.12-56 '7
PROGRAMLISTA
commodore 128 programbetoltes katalogusrol pattopont hasznalata
-\run2 '86.10-81 '3
PROGRAMLISTA
commodore 64 geos grafika geos szeruoperacios rendszer macintosh-tipusu pulldown menu input es info sablaK
-\run2 '86.11-90 '15
PROGRAMLISTA
commodore 64 grafika 50 uj utasitas beepitese -\run2 '86.12-80 '19
PROGRAMLISTA
commodore 64 grafika 640*400 pont kezelese pszeudo-3d basic bovites
-\run2 '86.12-124 '9
PROGRAMLISTA
commodore 64 grafika billentyuKkel behivhato uj basic utasitasoK ctrl-Q
-\run2 '86.11-86 '4
PROGRAMLISTA
commodore 64 jatekprogram robot rejtve -\happ '86.12-54 '4
PROGRAMLISTA
commodore 64 jatekprogram safe-cracker -\run2 '86.11-125 '3
PROGRAMLISTA
commodore 64 jatekprogram oKtatas/szin es hangsorozatoK emlekezetfejlesztesehez -\run2 '86.12-120 '4
PROGRAMLISTA
commodore 64 jatekprogram tenisz finaltag -\run2 '86.11-127 '6
PROGRAMLISTA
commodore 64 kodolas lemezedes basic rnd fuggveny felhasznalasa
-\run2 '86.10-116 '2

PROGRAMLISTA
commodore 64 lemezegyseg 1541 ellenorzo rutin a save with replace-hez
-\run2 '86.11-85 '1
PROGRAMLISTA
commodore 64 lemezegyseg 1541 floppykezelo basic bovites disk-basic v2.0 -\run2 '86.11-105 '7
PROGRAMLISTA
commodore 64 lemezegyseg 1541 uzemeltetesi trukKoK diskmonitor kezeles
-\run2 '86.10-96 '6
PROGRAMLISTA
commodore 64 lemezedes formatalas spur 37 -\happ '86.12-38 '3
PROGRAMLISTA
commodore 64 lista-formatalas
-\run2 '86.10-117 '5
PROGRAMLISTA
commodore 64 profi-ass/ram/rom reassemblalas -\run2 '86.10-87 '9
PROGRAMLISTA
commodore amiga grafika screencopy rutin -\run2 '86.11-66 '2
PROGRAMLISTA
sinclair spectrum grafika gepi rutin plusz ket also sor hasznalatahoz
-\happ '86.12-125 '1
PROGRAMLISTA
wizawrite nyomtatas illesztomodul utantoltese a wizaspell helyett
-\run2 '86.10-85 '2
PROGRAMLISTA
wordstar cp/m file-listazas alternativa a type helyett
-\happ '86.12-121 '1
PROGRAMOZAS
commodore 64 sys-utasitasoK hasznalata -\run2 '86.12-86 '2

Minden hétfőn 17-től 19 óráig

a Mikroszámítógép Magazin munkatársai és felkért szakértők válaszolnak az olvasók kérdéseire a szerkesztőségben: Budapest II., Fő u. 68. I. em. 109. vagy a 154-090 és a 154-250-es telefonon.

Minden kedden 17-től 20 óráig ENTERPRISE-klub a VSZM Közösségi Házban

(Bp. XI., Fehérvári út 120.)
Klubvezető: Romvári Gábor
Telefon: 450-950/473

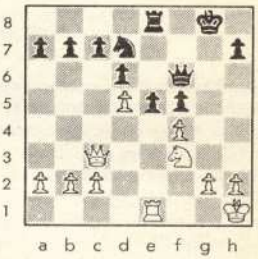
A játékfa és kiértékelése 4.

Az eddigiekben már láthattuk, hogy rendezett játékfa esetén — a randomszerűvel ellentétben — sokkal gyorsabb a fa kiértékelése. Ezért nagyon lényeges, hogy a fa felépítésénél milyen sorrendben vizsgáljuk egymás után a lépéseket. Programunk akkor lenne a leghatékonyabb, ha mindig az adott állásban megtehető legjobb lépéssel kezdenénk a számítást. Sajnos ezt nem tudjuk megtenni, hiszen nem ismerhetjük előre a legjobb lépést. Ha ismernénk, szükségtelen is lenne a játékfa felépítése és a fa végpontjaiban az állásértékelés. Ennek ellenére a legtöbb esetben heurisztikus alapon, kisebb-nagyobb hibával, de nagy valószínűséggel ki lehet választani a legjobbnak ígérkező lépéseket. A lényeg, hogy minél nagyobb fellépés mélységben rendezett a játékfa, annál gyorsabb lesz az értékelés.

Az elmúlt alkalommal bemutattam az iterációs eljárást, ami az esetek zömében hatékonyan működik és biztonságosan rendezi a fa közepét (de csak a közepét!). Ha például a harmadik féllépésben értékeljük az állást, akkor az iterációval elérjük, hogy az első és a második féllépésben nagy valószínűséggel a legjobb lépést vizsgáljuk elsőként, de az állásértékelés, esetünkben a 3. féllépés szintjén lehetséges lépések vizsgálati sorrendjére ennek az iterációs értékelésnek semmilyen hatása nincs. Tehát megállapíthatjuk, hogy az iterációs eljárás az alfa-vágás hatékonyságát elősegíti, de a végcsomópontokban levő nagyságrendekkel nagyobb számú béta-vágásra nincs hatással. Erre viszont különböző más típusú, heurisztikus módon működő, gyorsító hatású eljárásokat dolgoztak ki, amelyekben közös, hogy mindegyik a lépéslistán levő lépéseket rendezi valamilyen szempont szerinti sorrendbe. Az eljárások közötti legfőbb különbség a lépések előre rendezéséhez figyelembe vett tényezőkben rejlik.

A legtriviálisabb lépés-előrerendezés azon a kimutatáson alapszik, hogy a legtöbb állásban a pozícióértéket az ütések javítják. Ezek között is első helyen szerepel a visszaütés, illetve a legutoljára lépett figura ütése. Ha ezt az ütést fűzzük a lépéslista legelejére, elkerüljük, hogy a program az értelmetlen figuraáldozatokat sokáig vizsgálja, mert a visszaütés során — amit elsőként vizsgál — rögtön kiderül, hogy a többi válaszlépés megvizsgálása felesleges. Ezt tovább lehet finomítani azzal, hogy az olyan állásokban, ahol több lehetséges visszaütés kínálkozik, a program a legkisebb anyagi értékű figurával próbáljon elsőként ütni, és csak utána nézze anyagi érték szerinti sorrendben a többi bábuval történő visszaütést. Tehát elsőként gyaloggal, majd huszárral, futóval, bástyával, vezérrel, végül a királlyal próbálja meg az utoljára lépett ellenséges figura ütését.

Nézzük az *ábrán* látható hadállást! Amennyiben a világos Be1xe5 lépésére lehetséges visszaütések közül elsőként az e8 mezőn álló bástyával próbálkozna a program, abban az esetben kiegyenlítettnek tekintené az anyagi egyensúlyt, és tovább elemezné az állást. Vf6xe5-öt számítva pedig világos számára kimondottan előnyösnek tekintené a Be1xe5 lépést. Arra pedig, hogy ez a világos bástyával történt ütés kifejezetten rossz, csak a d6xe5 vagy a Hd7xe5 lépés számításánál jönne rá. Ha ezek a lépéslista végén szerepelnek, akkor az eddig megtett lépéseket — amelyek száma elég tetemes lehet — teljesen feleslegesen és értelmetlenül vizsgálta volna a program, és ez nagy időveszteséggel járna. A visszaütéseket a normális ütések követik a lépéslistán, szintén a figurák anyagi értékétől függő sorrendben, majd a „csendes" lépések következnek. A különleges lépéseket — az en passant lépést, a gyalogátváltozást, a sáncolást — szintén érdemes mintegy az ütések közé sorolva, az elsők között megvizsgálni.



A Kempelen Spectrum program fejlesztése közben lényegében az előbb említett módszerhez hasonló módon, de sokkal precízebben rendeztük a lépéslistát. Nemcsak azt vettük figyelembe, hogy milyen anyagi értékű figurával ütünk, hanem azt is, hogy az ellenfélnek melyik bábuját ütjük vele. Így például hiába üthetnénk az állásban gyaloggal egy gyalogot vagy huszárt, mégsem ez a lépés kerül a lépéslista elejére, hanem a huszárral, futóval történő vezérütés. Ezt úgy érjük el, hogy minden lépést pontozunk, és a pontszám alapján sorba rendezve írjuk azokat a listára. A módszerrel nemcsak az ütéseket lehet megfelelően rendezni, hanem a csendes lépéseket is. Ezek közül érdemes előre venni az aktív, támadó lépéseket, a centrumgyalog-lépéseket, végül a szélső gyalogok csendes lépéseit. A gyorsítás érdekében érdemes két lépéslistát kezelni: az egyikben az ütéseket, a másikban pedig a csendes lépéseket jegyezni. A csendes lépéseket csak akkor kell az előbb említett pontszám alapján rendezni, ha egyik ütő lépés sem vágja le a játékfát: ellenkező esetben felesleges volna. Mivel a nem ütő lépések vannak többségben, rendezésük az ütésekkel szemben legalább egy nagyságrendnyivel több időt emészt fel. Ezért csak akkor kell ezeket rendezni, ha valóban szükséges.

A lépéslista rendezéséhez az előbb említett szempontokon kívül a program felhasználhatja a már értékelt játékfarész által gyűjtött tapasztalatokat. Erre kézenfekvő megoldásnak mutatkozik, hogy azokat a lépéseket vizsgálja közvetlenül a visszaütő lépések után, amelyek már egyszer egy faágon ugyanilyen féllépés mélységben legjobbnak bizonyultak. A szakirodalom ezt nevezi gyilkos lépésnek (angol terminológia szerint killer move). Tehát a program minden szinten eltárol egy gyilkos lépést, vagyis a szintnek megfelelően az eddig legjobbnak találtat, és ezt az előbb említett módon felírja a lépéslistára, ha a játékfa felépítése közben ismét erre a szintre ér. Ha ez a lépés nem vágja le a fát, de akad egy másik vágó lépés, akkor az új lépés tekinthető gyilkosnak. Érdemes külön vizsgálni, hogy a legjobb lépés nem visszaütése. Ugyanis általában a visszaütés a fának egy másik ágán már nem legális, ezért ezt nem célszerű gyilkos lépésnek tekinteni.

Az alfa-béta algoritmusba ezt a *listán* bemutatott módon lehet beépíteni.

KOVÁCS P. ATTILA

```
AlfaBeta(p: pozicio; alfa,beta,melyseg: integer): integer
var
    legjobb,ertek,sz,t,m : integer;
    lepesek : array[1..MAX_SZELESSEG] of integer;
    gyilkos: array[1..MAX_MELYSEG] of integer;

begin
    if melyseg = 0 then  { terminalis csomopont? }
        return( Pontertek(p) );
    sz := Lepesgenerator( lepesek );
    if sz = 0 then       { nincs legalis lepes? }
        return( Pontertek(p) );

    ertek := -oo;
    for m := 1 to sz do
        begin
            t := -AlfaBeta( p.lepes[m], -beta,-alfa,melyseg-1 );

            if t > ertek then
                ertek := t;

            if ertek >= beta then
                begin
                  { levagja a fa tobbi reszet }
                  legjobb := p.lepes[m];
                  goto done;
                end;

            alfa := Max( alfa,ertek );
        end;
    done:
        gyilkos[melyseg-1] := legjobb;
        return( ertek );
end.
```

Aprócska játékos trükkök, melyek szórakoztatnak, és egy picit a matematikai ismereteket is felelevenítik

Petike adót számít

Apuka értetlenkedik, Petike meg hegyezi a fülét, ahogy ez a gyerekeknél szokás. Apuka nem érti, hogyan lehet egy adott képlettel — amit a munkahelyén mutattak neki — átszámítani a keresetét, amikor a jövedelemadó mértéke sávosan változik.

— Igaz, fel lehet írni egyenlet formájában az új bruttósított keresetemet, de nincs egyértelmű megoldása — morfondíroz apuka. Abból az elvből kiindulva, hogy a nyugdíjjárulék levonásával kézhez kapott jelenlegi fizetésemnek meg kell egyeznie az új adóval és a 10 százalék nyugdíjjárulékkal csökkentett 1988. évi jövedelmemmel, az egyenlőséget nem nagy dolog felírni. Az adóalap megállapításánál figyelembe veszem, hogy az évi béremből 12 000 forintot levonhatok:

$$BR - 0{,}01NY*BR = BU - 0{,}01SZJ(BU - SZH - 12\,000) - -SZK - 0{,}1BU$$

átrendezve:

$$BR - 0{,}01NY*BR = BU(1 - 0{,}1 - 0{,}01SZJ) + +0{,}01SZJ(SZH + 12\,000) - SZK$$

Az új bért kifejezve:

$$BU = \frac{BR(1 - 0{,}01NY) - 0{,}01SZJ(SZH - 12\,000) + SZK}{0{,}9 - 0{,}01SZJ}$$

ahol

BR — a régi bér (Ft)
BU — az új bér (Ft)
NY — az 1987-ben érvényes nyugdíjjárulék (százalékban)
SZJ — az adókulcs (százalékban)
SZH — a jövedelemsáv alsó határa (Ft)
SZK — a jövedelemsáv alsó határának ún. halmozott adója (Ft)

— Ezzel el is készültünk — mondta apuka —, de az egyenletnek nincs egyértelmű megoldása, mert az adósáv mértéke függ az új bértől, amelyet ki akarunk számolni. Előre nem tudjuk, hogy melyik sávba esik, ezért csak próbálgatással kaphatunk helyes eredményt.

Ekkor érkezett el Petike ideje. „Ezt a próbálkozást bízzuk a számítógépre!" felkiáltással hozzálátott a program megírásához.

A program megkérdezi az 1987. évi bért és a nyugdíjjárulék százalékát. Az adótáblázatot a program indításakor egy SZA nevű tömbbe olvassa be. A 130-as sorban levő feltétel a próbálgatások számát csökkenti, mivel az új bér nyilvánvalóan nem kerülhet alacsonyabb sávba, mint a bruttósítás előtti. A 180-as sor kiszámolja a közölt képlet szerint feltételezett adósávban az új bért. A 220-as sor a feltételezett teljes bérből kiindulva megállapítja a tényleges adót. Amennyiben megfelelő sávban számolt, a kétféleképpen megállapított nettó jövedelemnek meg kell egyeznie. Ellenkező esetben a program a következő sávban próbálkozik. A program szerkezetének legfőbb jellemzője ez az összehasonlítás.

A papa örömét csak fokozta, hogy Petike túl is teljesítette az eredeti feladatot. Ha már úgyis benne volt a munkában, kiegészítette a számítást azzal, amire valójában mindenki kíváncsi: mennyi teljes jövedelmet kell elérnie ahhoz, hogy bizonyos összeg maradjon a zsebében?

A program ezt hasonló elv szerint számolja ki, csak más képletet használ (lásd a 300-as sortól). A különböző forrásokból (más-más munkahelyekről, más-más tevékenységekből) származó jövedelmek megállapításánál nem veszi ugyan figyelembe a differenciáló tényezőket — mert ez manapság még határozatlan —, tehát gyakorlatilag az adóalapot számolja.

Az olvasó ne feledkezzen meg arról, hogy a programot nem az érvényes jogszabály szerint készítettük el, hiszen a cikk írásakor azt még nem is ismerhettük. Lényeg, hogy a program elve jól hasznosítható lépcsőzetes függvények értékeinek meghatározására.

A lista MS—BASIC-ben készült.

PINKE GYÖRGY

```
10 DIM SZA(11,4)
20 FOR I=1 TO 11
30 FOR J=1 TO 4
40 READ SZA(I,J)
50 NEXT J
60 NEXT I
70 DATA        0, 48000,  0,     0
71 DATA    48000, 70000,  20,    0
72 DATA    70000, 90000,  25, 4400
73 DATA    90000,120000,  31, 9400
74 DATA   120000,150000,  34,18700
75 DATA   150000,180000,  39,28900
76 DATA   180000,240000,  44,40600
77 DATA   240000,360000,  48,67000
78 DATA   360000,600000,  52,124600
79 DATA   600000,800000,  56,249400
80 DATA   800000,5000000, 60,361400
100 INPUT "KEREM A REGI BERT",BR
110 INPUT "KEREM A NYUGDIJJARULEK %-AT",NY
120 FOR I=1 TO 11
130 IF BR> SZA(I,2) THEN GOTO 150
140 K=I :GOTO 170
150 NEXT I
160 K=11
170 FOR I=K TO 11
180 BX=BR-.01*NY*BR-.01*SZA(I,3)*(12000+SZA(I,1))
185 BU=(BX+SZA(I,4))/(.9000001-.01*SZA(I,3))
190 P=I
200 FOR J=P TO 11
210 IF (BU-12000)>SZA(J,2) THEN GOTO 250
220 BX=BU-.1*BU-.01*SZA(J,3)*(BU-12000-SZA(J,1))
225 BP=BX-SZA(J,4)
226 BN=BR-.01*NY*BR
230 IF ABS(BN-BP)<10 THEN GOTO 270
250 NEXT J
260 NEXT I
270 PRINT "UJ BER:",BU
300 INPUT "AKAR TOVABB SZAMOLNI I/N",W$
310 IF W$="N" THEN GOTO 480
320 INPUT "NETTO JOVEDELEM?",N
330 FOR I=1 TO 11
340 IF N>SZA(I,2) THEN GOTO 360
350 K=I:GOTO 370
360 NEXT I
365 K=11
370 FOR I=K TO 11
380 BX=N-.01*SZA(I,3)*SZA(I,1)+SZA(I,4)
385 BJ=BX/(1-.01*SZA(I,3))
390 P=I
400 FOR J=P TO 11
410 IF BJ>SZA(I,2) THEN GOTO 450
420 NX=BJ-.01*SZA(J,3)*(BJ-SZA(J,1))
425 NP=(NX-SZA(J,4))
430 IF ABS(N-NP)<10 THEN GOTO 470
440 GOTO 460
450 NEXT J
460 NEXT I
470 PRINT "BRUTTO JOVEDELEM",BJ
480 END
```

Legfrissebb postánk sok érdekes információt, kérést, tanácsot, elmarasztalást és dicséretet tartalmaz. Tallózzunk együtt a levelek között!

Beke Gábor, Budapest,

Kővágó u. 4. 1108

Sok számát megvettem már lapjuknak, mert én is számítógép-tulajdonos vagyok. Az lenne a kérésem, hogy több Primo programot közöljenek.

A rossz Primo programtermésnek valószínűleg az az oka, hogy a gép gyártását megszüntették, és ráadásul ezeket a gépeket egyre kevesebb iskolában használják.

Gergely Miklós, Keszthely,

Zalka Máté út 1. 3/15. 8360

Az 1987/4. számunkban közölt Atari-ismertetőben figyeltem fel erre a gépre. A régit eladva és hozzágyűjtve, sikerült megvennem. A kezdeti öröm azonban hamar lelohadt, mert magnót a föld alól sem sikerült előkaparnom. Kérem, írják meg, hogy Atari 800XL-hez hol, mennyiért és mikor lehet magnót kapni. Azt írják, hogy bármely, hazánkban forgalmazott magnó kisebb átalakítással alkalmas Atarihoz. Írják meg ennek módját, anyagszükségletét is, és természetesen rajzot is küldjenek.

Az Önök lapja az egyetlen hazánkban, amelyhez ilyen kéréssel fordulhatok. Az egyetlen nívós számítástechnikai folyóiratnak tartom, de mégis van két nagy hibája. Észre kellene végre venni, hogy Magyarországon nemcsak Commodore-ok és Sinlcair gépek léteznek, és nem mindenki ért profi módon a programozáshoz. Én úgy vagyok az újsággal, hogy minden hónapban megveszem, elolvasom az utolsó betűig, majd átkozódni kezdek, hogy miért nincs nekem 80 ezer forintom C64-re vagy 128-ra.

Végül néhány ötlet, hogy milyen cikkeket látnék szívesen az Atariról: gépi kódolás, zene, grafika, sprite-technika a gépen, a játékvezérlő utasításokról bővebben stb. Mivel kezdő vagyok, leginkább ezek a dolgok érdekelnek, de ha nem találok anyagot az Atarira, nem tudok továbblépni. A BASIC-et ki lehet egyénileg kísérletezni, a gépi kódolást nem.

Úgy látszik, hogy nem lehet büntetlenül ismertetéseket közölni, hiszen a cikk hatására lecserélte számítógépét. Ami a magnóproblémáját illeti, azt tanácsolom, hogy keresse fel az NJSZT HCC Atari szekcióját: minden hónap első keddjén 17 órától van klubfoglalkozás az NJSZT-ben. Levelének másolatát egyébként elküldtem a szekció vezetőjének (Hirschler Tamás, telefon: 128-266), valamint a Skála-Coop Computer-Š boltjának és a Novotrade-Fotoelektronik szerviznek is. Miután egyre többeknek van ilyen gépük az országban, mi is igyekszünk Atari-írásokat és programokat közölni. Várjuk az Atari-hívő szerzők jelentkezését.

Fábián János, Gyál

Hunyadi u. 37. 2360

A Magazin 1987/5. számában egy Z80 szimulátorprogram leírására bukkantam. ZX-Spectrum gépemen elég sokat programozok, és én is úgy találom, hogy az assembler programok készítéséhez nagy segítséget ad ez a program. Kérem, írják meg, hogyan lehet hozzájutni, és van-e más felhasználói program is?

Az idézett cikk végén tett ígéretünknek megfelelően kérését a HCC Sinclair klubjához továbbítottuk.

Dr. Nagy András, Szeged,

Aradi vértanúk tere 3. 6720

Mostanáig próbáltam beszerezni a Magazin 1987/3. és 4. számát, de nem sikerült. Ezért Önt kérem, hogy küldje el ezeket a számokat. 1984-től vagyok a lap olvasója, és nem szeretném, ha ez a két szám hiányozna. Nagyon jó egyébként a 6. szám. Szinte csupa olyan cikk van benne, ami nagyon érdekel.

Az NJSZT titkárságán a következő régebbi számokat lehet még korlátozott példányban beszerezni: 1984/4—5., 1985/3—4—5., 1986 teljes évfolyam (számonként 24,— Ft), 1987/1—2—5—6—7. (számonként 30,— Ft).

Juhász László, Rém,

Szabadság u. 15. 6446

Plus/4 számítógép-tulajdonos tanuló vagyok. Nekem is elég sok gondot okoz egy nagyobb program betöltése a hosszú töltési idő miatt. Lapjuk múlt évi utolsó számában találtam egy C16 Turbo Tape programot. Szeretném megtudni, hogy ez használható-e az én gépemre is. Ha nem, akkor az Önök segítségével szeretnék turbóhoz jutni. Például úgy, hogy küldök kazettákat és Önök felveszik rá a programot, ha ez egyáltalán megoldható.

A C16 és Plus/4 egymással nem kompatibilis gépek, ezért nagy valószínűséggel az említett program nem fut a Plus/4-en. Sajnos nem tudjuk, hogy van-e Plus/4-es turbo program. Levelét azért közöljük, hogy hátha tud valaki segíteni. A szerkesztőség program másolására nem vállalkozik.

Füle Sándor, Kiskunhalas,

Sallai u. 18. 6400

Idézet az 1987/5. Magazinból: „. . . a következő számunkban már keresheti Áruk és áruk című új, állandó rovatunkat". Valóban kereshettem, nem szólt rám senki közben. De nem találtam.

A BNV-ről kezembe került egy áras TVC „katalógus". Szimpla floppy kb. 28 ezerért, az operációs rendszere még 7 ezer forint, összesen majdnem 35 ezer. Ez a gép árának csaknem háromszorosa. Józan paraszti eszem azt súgja, hogy ha az áraknak nem is, de az arányoknak hasonlítaniuk kellene a nyugatiakhoz.

Az igéretünk elhamarkodott volt, elnézését kérjük. Sajnos a VT és a nyugati szállítók technikai feltételei nem azonosak, ezért ne várjon hasonló árarányokat.

Rejtő Péter, Budapest

Darvas József u. 25. 1033

Időnként van szerencsém hozzájutni egy-egy rangosabb német magazinhoz. Természetesen ezekben is előfordulnak nyomdatechnikai vagy szerzői hibák, de a következő számban ott található, feltűnően kiemelve a korrekció. Ezt tisztességes módszernek tartom, és talán nem ártana követni a példát. Távol álljon tőlem, hogy szakembereket oktatni vagy bírálni akarjak, csak mindössze azt kívánom érzékeltetni, hogy a magamfajta „fogyasztó"-nak mik az elvárásai.

Remélem, észrevette, hogy a lapban sajnos többször is előforduló hibákat mi is szorgalmasan kijavítjuk, a javítást pedig közöljük. Sokszor az olvasók figyelmeztetnek bennünket egy-egy hibára. Ezeket a javításokat éppen ebben a rovatban szoktuk közzétenni.

Borz Andrea, Várpalota,

Tési út 27. 8100

Június 25-én a tv-híradóban egy riportra lettem figyelmes. Mikroelektronikai szakemberek képzéséről volt szó. A képzés szabadegyetem jellegű lenne, és felsőfokú képesítést ad. Amiben eltér az eddigiektől: nem kell felvételizni. A képzés teljesen önköltséges lenne. Sajnos a riportból nem derült ki, hogy *hol* lesz ez és *mikor* lehet rá jelentkezni.

A riport az LSI ATSZ-ről és a Gödöllői Agrártudományi Egyetem Mezőgazdasági Gépek szakának kezdeményezéséről szólt; az egyetemen mikroelektronikai nyitott tagozatot kívánnak létrehozni. Valószínűleg arról is hallott, hogy az NJSZT kezdeményezésére egy átfogóbb, nyílt egyetemi oktatási rendszert szervezünk. Úgy tervezzük, hogy ezt az új távtanulási technológiát alkalmazó intézményt több főhatóság alapítja. Számos egyetemmel, főiskolával, kutatóintézettel, oktató és vezetőképző központtal együttműködve alakítjuk ki a tananyagot és hozzuk létre a teljes oktatási rendszert. Terveink szerint az oktatás 1988-ban lépcsőzetesen indul, és 2-3 év alatt válik teljes körűvé. A mikor-ra és a hol-ra valószínűleg az év végén tudunk választ adni.

Turay Gábor, Budapest,

Kőrösi Csoma út 35—37. 1105

A Magazinban sok olvasótársammal együtt keveselltem a Spectrum programok mennyiségét. Ön ezekre a levelekre azt válaszolta, hogy az olvasók nem küldenek elég programot. Most én elküldöm Önnek az egyik programomat. A lista mellé cikket is írtam. Ha ezeket közlésre méltónak találja, szívesen látnám viszont a lapban.

A Magazin áprilisi számában közzétett levelemben felajánlottam Önnek, hogy hozzájárulnék a Spectrum-rovat bővítéséhez. Erre azonban Ön nem reagált. Most ismét felajánlom, hogy küldenék programokat, ötleteket, tippeket. Ha az itt mellékelt program nem üti meg a szintet, szívesen küldök másikat. Kérem, írja meg, hogy igényt tart-e programjaimra.

Köszönjük az első programot. Bizonyára észrevette, hogy sok olvasói programot és írást jelentetünk meg lapunkban, és szívesen fogadunk minden javaslatot, az Önét is, hogy például növeljük a Spectrum-cikkek számát. Azt azonban nem tudjuk megígérni, hogy minden munkája megjelenik. Egy-egy szám tartalmát a felelős szerkesztő állítja össze, aki a legjobb tudása szerint a legszínvonalasabb írásokat választja ki, és ezek közül is azokat, amelyek véleménye szerint az olvasókat a legjobban érdeklik. Ha sok írást küld, akkor előbb-utóbb valamelyiket biztosan közölni fogjuk. Különösen akkor, ha a programlistát friss festékszalaggal nyomtatja ki, ti. amit most küldött, azt egyáltalán nem tudjuk használni.

Az olvasók leveleit, írásait, programjait továbbra is szeretettel várja

KOVÁCS GYŐZŐ

Smith, B. R.:
Első könyvem a programozásról
(Budapest, 1987.
Műszaki Könyvkiadó,
48 oldal. Ára: 99,— Ft.)

A gyerekek számára a mikrók lassan már olyan természetesek, mint az autók. Szívesen látnak hozzá a programozáshoz is — és állíthatjuk, hogy meglepő eredményeket mutatnak fel. Egyszerű, látványos programból azonban igen kevés van forgalomban, főleg olyanokból, amelyek bármely gépen futtathatók. Ezt a könyvet viszont nemcsak a programok, hanem a magyarázó szövegek és a szemléletes ábrák, rajzok is varázslatossá teszik.

Link, W.:
BASIC a mérés-, a vezérlés- és a szabályozástechnikában
(Budapest, 1987.
Műszaki Könyvkiadó,
150 oldal. Ára: 69,— Ft.)

Ez a könyv elsősorban nem a szabályozás- és vezérléstechnika területén dolgozó szakembereknek szól, hanem olyan amatőröknek, akik saját személyi számítógépük alkalmazási területeit kívánják bővíteni ezekkel a lehetőségekkel. Bár a leírt kapcsolásokat a Tandy TRS 80 jelű (Radio Shack) géppel valósították meg, mégsem jelenthet túl nagy nehézséget a programokat más BASIC változatra átírni, vagy a kívánt interfészre módosítani.

Tudatosan csak egyedi kapcsolásokat ajánl a könyv, hogy a kezdőknek megkönnyítse az indulást a számítógéppel automatizált mérés-, vezérlés- és szabályozástechnika területén. Az már nem okozhat jelentős gondot, hogy több mérőkapcsolást egy automatizált mérőberendezéssé, illetve több vezérlőegységet egy bonyolult rendszerré kapcsoljon össze az olvasó.

Sebestyén Béla:
Helyi számítógép-hálózatok
(Budapest, 1987.
Műszaki Könyvkiadó,
378 oldal. Ára: 145,— Ft.)

Az elmúlt időszakban a számítástechnika és a távközléstechnika fejlődésével rohamos fejlődésnek indult a helyi hálózatok kiépítése.

A könyv mindazokat a hálózatokat ismerteti, amelyek helyi és egyidejűleg számítógépi jellegűek is. A sín- és gyűrűhálózatok változatain túl áttekintést ad a csillaghálózatokról, a fénykábelhálózatokról, a műszaki és a tudományos kutatásokban alkalmazott, illetve az ipari jellegű hálózatokról.

A kötet a hálózatok hardverleírásával foglalkozik, és nem tér ki a szoftver kérdésekre. Mivel a szerző — a helyi hálózatokra vonatkozó szűkös magyar nyelvű irodalom mellett — könyve megírásában elsősorban idegen nyelvű szakirodalomra támaszkodott, az olvasók tájékozódásának megkönnyítésére a terület angol nyelvű szakkifejezéseinek szótárát is összeállította. A kötet másik hasznos függelékében a leggyakrabban előforduló idegen nyelvű rövidítések feloldása található.

Heift:
CAD. Bevezetés a számítógéppel segített műszaki tervezésbe Commodore 64-es számítógéppel
(Budapest, 1987.
Data Becker—Novotrade,
164 oldal. Ára: 290,— Ft.)

A CAD a Computer Aided Design rövidítése, és számítógéppel segített műszaki tervezésnek vagy automatizált műszaki tervezésnek (AMT) lehet fordítani. Olyan mérnöki tevékenységek sorozata, amelyekben a tervezéshez, a számítás—méretezéshez, a műszaki rajz előállításához, a dokumentáláshoz és a szimulációhoz, teszteléshez a számítógépet használhatjuk.

Az a konstruktőr, aki CAD programokat használ, a fiókba rakhatja a ceruzáját, vonalzóját és körzőjét. Helyettük valamilyen beviteli eszközt, illetve a számítógép billentyűzetét használhatja, mivel minden a képernyőn és a rajzgépen valósul meg.

A könyv első fejezete a számítógéppel segített műszaki tervezés alapelveiről szól, különösen a C64-re figyelemmel. A szerző a második fejezetben mutatja be azokat a legfontosabb programokat, amelyekre a műszaki tervezési feladatok megoldásához van szükség.

A harmadik fejezet ismerteti, hogy milyen tágabb lehetőségei vannak a számítógéppel segített műszaki tervezésnek. A negyedik fejezetben olyan programelemeket mutat be, amelyek feltétlenül szükségesek a műszaki tervek ábrázolásához. A műszaki tervezési feladatok megoldásához egy CAD rendszert is ismertet: a CADDY-MAT-ot.

A könyv nem tartalmaz kész rendszert. Sokkal inkább azt kívánja bemutatni, hogyan lehet különböző programokat menü segítségével párbeszédes üzemmódban alkalmazni.

Barakonyi Károly:
Táblázatkezelő rendszerek (Lotus 1. 2. 3., Framework, CalcStar, Multiplan, CalcResult, CalQula)
(Budapest, 1987.
LSI ATSZ,
554 oldal. Ára: 361,— Ft.)

A számítástechnika fejlődésében minőségileg új lépcsőt jelentenek a felhasználó által közvetlenül alkalmazható, nagy teljesítményre képes integrált programcsomagok. Ezek között különösen népszerűek a táblázatok kezelésére kifejlesztett rendszerek. Különösebb számítástechnikai tudást nem igényelnek, segítségükkel a témát legjobban ismerő szakember, a felhasználó (mérnök, közgazdász, ügyintéző) akár saját maga építheti fel a feladatot, melynek megoldását interaktív módon követheti végig.

Ezeknek a rendszereknek a terjedését gátolja, hogy többségük idegen nyelvű dokumentációval érkezik hazánkba, fordításuk késik, széles körben nem érhető el. A gyári dokumentációk terjedelmesek, használatuk időigényes. Problémát jelent a feladathoz leginkább illeszkedő rendszer kiválasztása is.

A kézikönyv célja, hogy a legnépszerűbb táblázatkezelő rendszerekről magyar nyelvű leírást adjon, amely vizsgálja az erős és a gyenge pontokat, a javasolt alkalmazási területeket is. Az angolul nem tudók igényeit kielégítve tárgyalja a műveleteket, a parancsokat, az üzeneteket, a kulcsszavakra kiejtési és jelentési szótárt ad. A kezdő felhasználóról minimális gépkezelői ismeretet tételez fel. A számítástechnikában gyakorlatlan alkalmazót nagyszámú szövegközi példával, a gyakorlott felhasználót a finomabb megoldások részletkérdéseinek tárgyalásával segíti.

Robotokkal

Az IBM ipari robotok alkalmazásával kívánja árait csökkenteni, ezért a Kentucky-beli Lexingtonban teljesen automatizált gyártósoros írógép-összeszerelő üzemet létesített 350 millió dolláros költséggel.

Az eddigi 4 órával szemben alig egy óra alatt készül el egy írógép, az éves termelést megduplázták, miközben a létszámot 6 ezerről 2 ezerre csökkentették. Az itt készülő elektronikus írógépek jobb minőségűek, megbízhatóbbak, áruk 2800 helyett csak 1800 dollár. A robotokat irányító számítógépek pillanatok alatt átprogramozhatók, s a robotok bármit összetudnak rakni, ami 50 × 55 × 45 cm-nél kisebb, és felülről összeszerelhető.

A robotosítás árnyoldalai közé tartozik, hogy a telepen évente legalább 1 millió egységet kell összeszerelni ahhoz, hogy a termelés ne legyen veszteséges; ezért például más üzemekből ide helyezték át a számítógép billentyűzetének összeszerelését.

A robotosítás megkövetelte a teljes gyártási szerkezet átalakítását. Az új típusnak csupán 900 alkatrésze van (a régebbi típusoknak háromszor ennyi volt), és csak 11 olyan alkatrészt tartalmaz, amely más termékekben nem szerepel. Az alkatrészraktár csak 5-6 napos készletet tárol, ugyanakkor az alvállalkozók számát is radikálisan csökkentették. Az eddigi 700 helyett jelenleg 100 szállít, de csak 60 megtartását tervezik.

Kismamák, figyelem! Váratlan veszélyek

Azok a terhes nők, akik szabályozott klímájú és rendkívül pormentes levegőjű helyiségekben, az úgynevezett tisztaszobákban dolgoznak a mikroelektronikai elemgyártáson, s eközben mérgező vegyületekkel maratják az áramköröket, erősen kockáztatják, hogy vetélés folytán elveszítik gyermeküket. Több más amerikai mikroelektronikai elemgyár után az AT and T is áthelyezett átmenetileg más munkahelyre 15 terhes nőt. A DEC cég már előzőleg megbízást adott egy tanulmányra, melynek megállapítása: az ilyen extrasteril körülmények között dolgozó terhes nők 39%-a vetél el (az átlag 20%).

Háziújságok

Az Egyesült Államokban néhány vállalkozó szellemű újságíró felismerte, hogy a lézernyomtatós számítógép-konfigurációk felfoghatók házi nyomdaként is. Így aztán ma már gomba módra szaporodnak a házi készítésű újságok. Robert Jacobson fairbanksi gitártanár lapja, a Fairbanks Music and Entertainment példányszáma nyolc hónap alatt 19 ezerre emelkedett, s a lap egy-egy száma ötven centbe kerül. Susan Ovan 14 ezer dollárt fektetett be: vett három Macintosh számítógépet és egy hatezer dolláros lézernyomtatót: így alapította meg a Hull Newsweekly című lapját. Tom Hamilton – egy Macintosh és egy Apple lézernyomtató tulajdonosaként – a Balloon Life című lapot jelenteti meg 1700 példányban. A költségek csökkenése ennél a lapnál a legfeltűnőbb. A Balloon Life egy-egy oldalának az előállítási költsége lézernyomtatón mindössze 60 centbe kerül, míg ugyanez hagyományos technikával 4-8 dollár lenne.

A házi újságok szerkesztőségeiben egyébként szinte kizárólagos szerephez jutottak a rendkívül barátságos, széles körű grafikus lehetőségeket biztosító szoftverrel rendelkező Macintosh gépek.

NSZK-vívmány: műanyag igazolvány

Az NSZK-ban az idén új típusú, számítógépes leolvasásra is alkalmas személyi igazolványt vezettek be. A jövőben a hagyományos, papírból készült okmányt fényképpel ellátott műanyag kártya váltja fel: az egyik oldalán tulajdonosának fényképe, aláírása, születési adatai s az érvényesség tartama áll, míg a másik oldalon lakcíme és személyleírása, azaz termete, szeme színe stb.

Az igazolvánnyal való mindenféle visszaélés, hamisítás kizárt, fizikai adottságai miatt pedig nem rongálódik. Mindezek ellenére ennek az európai újdonságnak számító megoldásnak a bevezetése belpolitikai viták keresztüzében áll. A folyamat azonban visszafordíthatatlan. Jövőre már megkezdik az NSZK-beli útlevelek kicserélését is, hasonlóképpen műanyag kártyára.

A munkaidőalap hathatós védelme: bejárás munkaidőben

A nagy svéd elektronikai iparvállalat, az ASEA székhelye Västeras városa. A Stockholmban lakó és naponta vasúton ingázó dolgozóinak a cég egy átalakított vasúti kocsiban több mint 20 munkahelyet (személyi számítógépekkel, írógépekkel), valamint tárgyalótermet és társalgót rendezett be. A munkahelyeket telefonokkal is felszerelték, a vonalak az NMT mozgó telefonrendszerhez kapcsolódnak. A cég itt is dolgozó alkalmazottainak a napi utazási idő most már természetesen munkaidőnek számít.

Mérgezés

A balassagyarmati kórház főgyógyszerésze, dr. Márton György a számítógépet hívta segítségül, hogy a toxikológus orvosok munkáját megkönnyítse. A forgalomban lévő növényvédő szereket gyors felismerésük érdekében különféle jellemzőik alapján csoportosította: szín, halmazállapot, csomagolás, felhasználás, hatásmechanizmus stb. szerint. A program jelenleg 720 vegyszert tartalmaz, de mindig kiegészítik az újólag piacra kerülőkkel. A keresés interaktív: a válaszok alapján szűkül a vizsgált kör, végül meghatározható a keresett vegyszer neve. A mérgező vegyszerek mellett természetesen mindig szerepel az ellenanyag is.

A program máris életeket mentett. Több, a miskolci megyei kórházba mérgezéses tünetekkel beszállított betegnél állapították meg segítségével a véletlenül megivott vegyszer nevét, s ennek köszönhetően a detoxikáló beavatkozás sikerült.

Dombor-tévé

Ijesztő gyorsasággal nő a született vakok száma. A civilizált országokban azért, mert az igen kis súlyú koraszülöttet is életben tudják tartani, de ezzel nő annak a kockázata, hogy az ilyen gyermek esetleg vak lesz. A fejletlen országokban a trachoma, a rossz táplálkozás, a vitaminhiány miatt vesztik el sokan a szemük világát.

Pontosan ezért fokozódik minden olyan eszköz jelentősége, amely javítja az életvitelüket, könnyíti a társadalmi beilleszkedésüket.

Kruppa László és Verő András ötlete a következő. Egy közönséges videokamera képét átalakítják térben kitapintható képpé. A térbeni képet négyezer mozgó tüske teszi érzékelhetővé. A berendezés előnye, hogy nemcsak nyomtatott, hanem mindenféle írott szöveget, képet, rajzot is „láthatóvá" tesz. Aki látott már műanyagba domborított térképet, az jól el tudja képzelni, mit nyújt a dombor-tévének keresztelt berendezés. A dombor-tévét perifériaként egy professzionális mikroszámítógép működteti.

Az egyelőre még csak tervekben megalkotott dombor-tévé prototípusának kifejlesztése 15-16 millió forint. Jelenleg folyik a fejlesztő társak „toborzása": ha beszállnának a – főképpen nyugat felé orientált – üzleti vállalkozássá fejlesztett tervbe, ennek fejében különféle előnyöket kapnának az értékesítés során.

VÉGRE MEGJELENTEK

AZ OMFB által támogatott IBM XT, AT kompatibilis olcsó számítógépeink.

MEGRENDELHETŐK

korlátozott darabszámban bemutatótermünkben:
1075 Budapest, Majakovszkij u. 1/D
Levélcím: 1475 Budapest, Pf.: 225
Telefon: 221-623
Telex: 22-7734

Vállaljuk
OLCSÓ GÉPEINKBŐL
lokális hálózatok tervezését,
telepítését
MEGRENDELŐINK
IGÉNYE szerinti
konfigurációban.

Keresse fel bemutatótermünket,
ahol információval, tanácsadással,
számítógép-bemutatóval
állnak munkatársaink
reggel 9-től délután 17 óráig
az érdeklődők
rendelkezésére.

MŰSZERTECHNIKA
KISSZÖVETKEZET

Budapest, Szállás u. 21. 1107
Levélcím: Pf. 225. 1475
Bemutatóterem: Majakovszkij u. 1/D 1075

Telefon: 471-590
Telex: 22-7734
Telefon: 221-623

Egyedi és általános szoftverek a vállalati ügyvitel területén. Termelésirányítási, raktárgazdálkodási programcsomagok egyedi és hálózatos gépeken.

A szoftvereket kívánság szerint, megrendelőink igényeinek és a helyi viszonyoknak megfelelően átalakítjuk, adaptáljuk. Helyzetfeltárással, rendszertervezéssel, betanítással, a megfelelő számítógépes konfiguráció kiválasztásával és beszerzésével segítünk vállalati problémáik megoldásában.

A garanciális idő letelte után átalánydíjas szervizszolgáltatást vállalunk számítógépeikre.

Egyedi fejlesztések, tervezések és témaorientált feladatok megoldásával is forduljanak hozzánk.

µS MIKROSZERVÍZ Számítástechnikai Műszaki Fejlesztő
1141 Budapest, Kőszeg u. 4. Kisszövetkezet
Szerviz: 1141 Budapest, Kőszeg u. 4. Tel.: 831-805
Iroda: 1136 Budapest, Sallai u. 36. Tel.: 200-685
Telex: 22-7946